Análisis y evaluación de vulnerabilidades en infraestructura de seguridad de un centro de datos de proveedor de Internet (ISP)

Carlos Holmes Fernández Rivera

Asesor

Yenny Stella Núñez Álvarez

Universidad Nacional Abierta y a Distancia UNAD

Escuela De Ciencias Básicas, Tecnología e Ingeniería – ECBTI

Especialización en Seguridad Informática

2026

Dedicatoria

A mis padres, Aida Alicia Rivera y Carlos Raúl Fernández Bonilla, quienes con su ejemplo de esfuerzo y resiliencia me enseñaron que los obstáculos no definen el camino, sino la fortaleza para recorrerlo.

A mis hermanos, Whilmer, Milena y Néstor, por ser mi soporte inquebrantable y la alegría constante en la travesía.

Este logro es tan mío como suyo.

Resumen

El presente proyecto de grado tiene como objetivo identificar posibles vulnerabilidades en la infraestructura de seguridad de una empresa proveedora de servicios de Internet (ISP), documentar los hallazgos obtenidos y proponer planes de acción que contribuyan al fortalecimiento de las políticas de seguridad informática. Estas acciones buscan mitigar riesgos y proteger los activos críticos de la empresa, garantizando así la integridad, disponibilidad y confidencialidad de la información.

La implementación del proyecto se llevará a cabo mediante la técnica de pruebas de penetración de tipo Caja Blanca, la cual permite evaluar el sistema con conocimiento previo de su arquitectura interna. La metodología principal adoptada será PTES (Penetration Testing Execution Standard), complementada con las buenas prácticas definidas en los marcos normativos NIST SP 800-115 y OSSTMM (Open Source Security Testing Methodology Manual), con el fin de asegurar un enfoque integral y estructurado.

El proyecto tendrá una duración estimada de 10 meses, durante los cuales se realizará un levantamiento detallado de la infraestructura lógica y física de la red, así como la preparación de los entornos de prueba y la configuración de herramientas especializadas. Posteriormente, se ejecutará la identificación de vulnerabilidades mediante técnicas de análisis estático y dinámico, finalizando con la documentación exhaustiva de los hallazgos y la formulación de recomendaciones concretas y viables para la mejora de la postura de seguridad de la organización.

Palabras clave: Seguridad informática, pruebas de penetración, Caja Blanca, PTES, NIST SP 800-115, OSSTMM, infraestructura crítica, ciberseguridad, vulnerabilidades.

Abstract

This undergraduate project aims to identify potential vulnerabilities in the security infrastructure of an Internet Service Provider (ISP), document the findings, and propose action plans to strengthen the company's cybersecurity policies. These actions seek to mitigate risks and protect the company's critical assets, thereby ensuring the integrity, availability, and confidentiality of information.

The project will be implemented using the White Box penetration testing technique, which allows for system evaluation with prior knowledge of its internal architecture. The main methodology adopted will be PTES (Penetration Testing Execution Standard), complemented by the best practices defined in the NIST SP 800-115 and OSSTMM (Open Source Security Testing Methodology Manual) regulatory frameworks, to ensure a comprehensive and structured approach.

The project will have an estimated duration of 10 months. During this time, a detailed survey of the logical and physical network infrastructure will be conducted, along with the preparation of test environments and the configuration of specialized tools. Subsequently, vulnerability identification will be executed using static and dynamic analysis techniques, concluding with the thorough documentation of the findings and the formulation of concrete and feasible recommendations for improving the organization's security posture.

Keywords: Information security, penetration testing, White Box, PTES, NIST SP 800-115, OSSTMM, critical infrastructure, cybersecurity, vulnerabilities.

Tabla de Contenido

Introducción 17

Justificación 19

Definición del Problema 21

Antecedentes del Problema 21

Formulación del Problema 23

Objetivos 24

Objetivo General 24

Objetivos Específicos 24

Marco Referencial 25

Marco Conceptual 25

Diseño de Metodología 30

Metodología PTES 32

Fase Interacciones Previas 32

Recopilación de la Información 32

Modelado de Amenazas 34

Análisis de Vulnerabilidades 34

Explotación 35

Post-Explotación 35

Reporte 36

Desarrollo de Los Objetivos 37

Establecer los Activos de la Infraestructura de Seguridad 37

Fase Interacciones Previas 37

Recopilación de la Información 39

Recopilación Pasiva de la Información 39

Clasificación de Activos 42

Recopilación Semi-Pasivo de la Información 47

Herramientas y Bibliotecas Utilizadas 47

Preparación del Entorno 48

Desarrollo del Script Python. 53

Ejecución del script. 57

Recopilación Activa de la Información 61

Técnica de Escaneo 61

Instalación de Scapy 65

Escaneo de Host en Puertos Específicos 67

Escaneo de Puertos 75

Escaneo de Servicios 80

Modelado de Amenazas 85

Metodología 85

Recolección de la Información 86

Identificación y Clasificación de los Activos 86

Identificación de la Comunidad de Amenazas 86

Correlación de Amenazas 87

Mitigaciones a Amenazas 97

Modelo STRIDE 99

Nota. La presente tabla muestra la descripción del modelo STRIDE. 100

Análisis de STRIDE sobre Activos Críticos 100

Identificar las Vulnerabilidades en la Infraestructura 101

Herramienta de Escaneo Nessus. 101

Principales Características de Nessus 102

Creación de Políticas de Escaneo 103

Vulnerabilidades Encontradas 104

Resumen de Riesgo 104

Top 10 de Vulnerabilidades Encontradas 105

Top 10 de Host con más Vulnerabilidades 107

Documentar Hallazgos Obtenidos del Análisis de Vulnerabilidades 108

Vulnerabilidades por Protocolo 109

Resumen de Vulnerabilidades Encontradas 110

Detalle de Vulnerabilidades por Activo 114

Medidas y acciones para fortalecer la seguridad de la infraestructura del centro de datos de la compañía 148

Endurecimiento de la Seguridad 149

Auditorías Periódicas y Pruebas de Penetración 149

Conclusiones 150

Recomendaciones 152

Apéndices 156

Lista de Tablas

Tabla 1 *Levantamiento de la Información* 40

Tabla 2 *Activos para Evaluar* 41

Tabla 3 *Criticidad en la Continuidad del Negocio* 43

Tabla 4 *Criticidad Exposición de Amenazas* 43

Tabla 5 *Criticidad en la Información* 44

Tabla 6 *Clasificación de Activo Según Criticidad* 45

Tabla 7 *Parámetros del Script* 55

Tabla 8 *Vulnerabilidades Encontradas Recopilación Semi-Pasiva* 59

Tabla 9 *Comparación Herramientas de Escaneo de Red* 64

Tabla 10 *Librerías Implementadas en Script Recopilación Activa* 68

Tabla 11 *Listado de Activos Encontrados en Puertos (80, 4443, 8443, 23)* 75

Tabla 12 *Resultado de Puertos Descubiertos en Activos de Red* 79

Tabla 13 *Características Del Método Get_Banner* 80

Tabla 14 *Resultados del Script Recopilación Activa* 84

Tabla 15 *Metodología para Modelado de Amenazas* 85

Tabla 16 *Clasificación de los Activos* 86

Tabla 17 *Clasificación de Amenazas* 86

Tabla 18 *Resultados Correlación de Amenazas* 87

Tabla 19 *Resultado Interacción SWs y Telnet* 88

Tabla 20 *Resultado de Interacción VPN* 89

Tabla 21 *Resultado Interacción Protocolo HTTP* 90

Tabla 22 *Resultado Interacción Protocolo HTTPS* 91

Tabla 23 *Resultado Interacción OSINT Pasivo* 92

Tabla 24 *Resultado Interacción OSINT Pasivo* 93

Tabla 25 *Resultado Interacción Protocolo SSH* 94

Tabla 26 *Resultado Interacción Filtrado Tráfico* 94

Tabla 27 *Resultado Interacción LAN* 96

Tabla 28 *Resultado Interacción NAS* 96

Tabla 29 *Resultado Interacción WAN* 97

Tabla 30 *Modelo STRIDE* 100

Tabla 31 *Análisis STRIDE sobre Activos* 100

Tabla 32 *Definición de Políticas de Escaneo* 103

Tabla 33 *Vulnerabilidades Encontradas* 104

Tabla 34 *Criticidad de Vulnerabilidades por Activo* 108

Tabla 35 *Resumen de Vulnerabilidades CVE* 110

Tabla 36 *Unencrypted Telnet Server* 118

Tabla 37 *SSL Certificate Cannot Be Trusted* 119

Tabla 38 *Huawei Campus Switch DoS (HWPSIRT-2014-0112)* 121

Tabla 39 *Huawei eSap Platform DoS (HWPSIRT-2014-0111).* 122

Tabla 40 *SNMP Getbulk Large Max-Repetitions Remote DoS* 122

Tabla 41 *Huawei Campus Series Switches Remote Buffer Overflow DoS (HWPSIRT-2015-02014)* 123

Tabla 42 *Huawei Campus Switch Multiple Vulnerabilities (HWPSIRT-2014-0315 - HWPSIRT-2014-0318)* 124

Tabla 43 *SSH Username Information Disclosure Vulnerability in Huawei Campus Series Switches*..... 125

Tabla 44 *PfSense < 2.2.3 Multiple Vulnerabilities (SA-15_07) (Logjam)* 126

Tabla 45 *PfSense < 2.3.3 Multiple Vulnerabilities (SA-17_01 - SA-17_03)* 127

Tabla 46 *PfSense < 2.3.5 Multiple Vulnerabilities (KRACK)* 128

Tabla 47 *PfSense < 2.4.3 Multiple Vulnerabilities (SA-18_01 / SA-18_02 / SA-18_03) (Meltdown) (Spectre)* 129

Tabla 48 *PfSense < 2.4.5 Multiple Vulnerabilities* 130

Tabla 49 *PfSense < 2.2.6 Multiple Vulnerabilities (SA-15_09 / SA-15_10 / SA-15_11)*..... 131

Tabla 50 *Unix Operating System Unsupported Version Detection* 131

Tabla 51 *PfSense Unsupported Version Detection*..... 132

Tabla 52 *PfSense < 2.3.1-p1 Multiple Vulnerabilities (SA-16_05)*..... 133

Tabla 53 *PfSense < 2.3.1-p5 Multiple Vulnerabilities (SA-16_07 / SA-16_08)*..... 134

Tabla 54 *PfSense < 2.3.1 Multiple Vulnerabilities (SA-16_03 / SA-16-04)*..... 135

Tabla 55 *Network Time Protocol Daemon (ntpd) Read_Mru_List() Remote DoS*..... 136

Tabla 56 *PfSense < 2.2.4 Multiple Vulnerabilities (SA-15_07)* 137

Tabla 57 *SSL Self-Signed Certificate.* 138

Tabla 58 *TLS Version 1.0 Protocol Detection, TLS Version 1.1 Deprecated Protocol* 139

Tabla 59 *PfSense < 2.4.2 Multiple Vulnerabilities (SA-17_07)* 140

Tabla 60 *JQuery 1.2 < 3.5.0 Multiple XSS* 141

Tabla 61 *OpenSSL AES-NI Padding Oracle MitM Information Disclosure.* 142

Tabla 62 *SSH Terrapin Prefix Truncation Weakness (CVE-2023-48795)*..... 143

Tabla 63 *Network Time Protocol (NTP) Mode 6 Scanner* 144

Tabla 64 *SSL Certificate Expiry.* 145

Tabla 65 *SNMP 'GETBULK' Reflection DDoS.* 146

Tabla 66 *Web Server Allows Password Auto-Completion.* 147

Tabla 67 *Acciones para Fortalecer la Seguridad en la Infraestructura* 148

Lista de Figuras

Figura 1 *Topología de Infraestructura de Seguridad* 46

Figura 2 *Preparación del Entorno para el Análisis de Vulnerabilidades.* 49

Figura 3 *Entorno Virtual Kali Linux.* 50

Figura 4 *Descarga del Paquete Miniconda.* 51

Figura 5 *Preparación del Entorno Virtual en Python.* 51

Figura 6 *Instalación de la Herramienta Shodan.* 52

Figura 7 *Instalación del Paquete Dotenv.* 53

Figura 8 *Interfaz Web de la Herramienta Shodan.* 53

Figura 9 *Librerías Necesarias para Construcción del Script.* 55

Figura 10 *Script de Automatización para Búsqueda de Activos Expuesto a Internet.* 56

Figura 11 *Script de Automatización para Búsqueda de Activos Expuestos en Internet.* 56

Figura 12 *Resultados de la Búsqueda en Shodan con el Script de Automatización.* 57

Figura 13 *Interfaz Web de Firewall Expuesto a Internet.* 58

Figura 14 *Proceso de Conexión Protocolo TCP* 62

Figura 15 *Instalación de Librería Scapy* 66

Figura 16 *Instalación de Librería TDDM para Añadir Barra de Progreso Durante Ejecución del Script.* 66

Figura 17 *Instalación de Librería Rich, para Presentación de Datos en Tablas.* 67

Figura 18 *Librerías Necesarias para Ejecución del Script Escaneo de Host.* 68

Figura 19 *Desactivación del Registrador Automático de la Librería Scapy.* 69

Figura 20 *Definición de la Clase Principal del Script.* 69

Figura 21 *Definición del Constructor de Clase del Script.* 70

Figura 22 *Configuración de los Parámetros que Recibe la Librería Scapy para el Escaneo.* ... 70

Figura 23 *Personalización del Paquete TCP.* ... 71

Figura 24 *Código para Leer Lista Especifica de los Activos a Escanear.* ... 71

Figura 25 *Configuración de la Librería Tqdm en el Script.* ... 72

Figura 26 *Configuración de la Librería Pretty para la Presentación de Resultados.* ... 72

Figura 27 *Configuración de Parámetros del Script Main.* ... 73

Figura 28 *Flujo de Trabajo Recopilación Activa.* ... 73

Figura 29 *Resultados del Script Escaneo de Host en Puertos Específicos.* ... 74

Figura 30 *Integración del Módulo Socket al Script.* ... 76

Figura 31 *Integración del Método Port Scan al Script.* ... 77

Figura 32 *Flujo de Trabajo Modulo Escaneo de Puertos.* ... 78

Figura 33 *Resultados de Escaneo de Puertos.* ... 79

Figura 34 *Integración del Módulo Get Banner al Script.* ... 80

Figura 35 *Integración del Módulo Service Scan al Script.* ... 81

Figura 36 *Flujo de Trabajo Módulo Escaneo de Servicios.* ... 82

Figura 37 *Resultado de Ejecución de Script para Obtener Banners.* ... 83

Figura 38 *Diagrama Modelado de Amenazas.* ... 87

Figura 39 *Resultado Interacción SWs, Telnet.* ... 88

Figura 40 *Resultado Interacción VPNs* ... 89

Figura 41 *Resultado Interacción Protocolo HTTP.* ... 90

Figura 42 *Resultado Interacción Protocolo HTTPS* ... 91

Figura 43 *Resultado Interacción OSINT Pasivo.* ... 92

Figura 44 *Resultado Interacción OSINT Pasivo* ... 93

Figura 45 *Resultado Interacción Protocolo SSH.* 94

Figura 46 *Resultado Interacción Tráfico Filtrado.* 94

Figura 47 *Resultado Interacción Tráfico LAN.* 95

Figura 48 *Resultado Interacción Servidores.* 96

Figura 49 *Resultado Interacción WAN.* 97

Figura 50 *Interfaz Web Software Nessus* 101

Figura 51 *Creación de Política de Escaneo en Nessus.* 104

Figura 52 *Resumen de Vulnerabilidades de Acuerdo con su Riesgo.* 105

Figura 53 *Top de Vulnerabilidades Encontradas* 106

Figura 54 *Top de Activos con más Vulnerabilidades* 107

Figura 55 *Vulnerabilidades Encontradas por Protocolo.* 109

Figura 56 *Activos con Mayores Vulnerabilidades Descubiertas.* 110

Figura 57 *Escaneo 192.16x.xx.xx* 114

Figura 58 *Escaneo 192.16x.xx.xxx* 114

Figura 59 *Escaneo 192.16x.xxx.xxx* 115

Figura 60 *Escaneo 192.xx.xx.xxx* 115

Figura 61 *Escaneo 192.16x.xx.xxx* 116

Figura 62 *Escaneo 192.16x.xx.xx* 116

Figura 63 *Escaneo 2x0.2xx.xx.xxx* 116

Figura 64 *Escaneo 1xx.xx.xx.xxx* 117

Figura 65 *ICMP Timestamp Request Remote Date Disclosure* 117

Figura 66 *Escaneo 192.1xx.xxx.xx* 119

Figura 67 *Escaneo 192.1xx.xx.xx* 121

Figura 68 *Escaneo 2x0.2xx.xx.xx* 126

Figura 69 *Escaneo 2x0.2xx.xx.xxx* 145

Lista de Apéndices

Apendice A *Script Constructor de Clase* 156

Apéndice B *Configuración de Parámetros del Script Main* 161

Apéndice C *Glosario……162*

Introducción

En la era de la cuarta revolución industrial, caracterizada por la digitalización masiva y la interconexión global, las organizaciones enfrentan un panorama de ciberseguridad cada vez más complicado y desafiante. Los ataques cibernéticos han evolucionado en sofisticación, frecuencia e impacto, poniendo en riesgo no solo la información sino también la continuidad operativa de empresas e instituciones. En este contexto, los datos se han consolidado como uno de los activos más valiosos, lo que convierte su protección en una prioridad estratégica para cualquier organización.

Los centros de datos, como núcleos críticos que albergan servicios esenciales y grandes volúmenes de información sensible, son objetivos recurrentes de amenazas externas e internas. Garantizar su seguridad perimetral no solo salvaguarda la integridad y confidencialidad de los datos corporativos, sino que también preserva la confianza de clientes, socios y stakeholders. Un incidente de seguridad en estas infraestructuras puede resultar en pérdidas económicas, daños reputacionales e incluso consecuencias legales, especialmente en sectores altamente regulados.

Este proyecto de grado se centra en evaluar la infraestructura de seguridad del centro de datos de una empresa proveedora de servicios de Internet (ISP), con el objetivo de identificar vulnerabilidades potenciales y proponer planes de mejora alineados con las mejores prácticas y estándares internacionales en ciberseguridad. Para ello, se emplea un enfoque metodológico riguroso, combinando técnicas de pentesting de Caja Blanca con herramientas especializadas como Nessus, Scapy y Shodan, así como marcos de referencia reconocidos como PTES, NIST SP 800-115 y OSSTMM.

La importancia del presente trabajo radica en su contribución al fortalecimiento de los mecanismos de defensa para la empresa proveedora de servicios de Internet (ISP). ante un

entorno de amenazas en constante evolución. Los hallazgos y recomendaciones derivados del análisis no solo permitirán mitigar riesgos inmediatos, sino también establecer un marco de seguridad proactivo y resiliente, en línea con estándares como ISO/IEC 27001 y las normativas nacionales en materia de seguridad digital. Además, este proyecto sirve como modelo aplicable a otros centros de datos que enfrenten desafíos similares, destacando la importancia de la evaluación continua y la adaptación a las nuevas tendencias en ciberseguridad.

Justificación

La auditoría de seguridad informática es una herramienta fundamental para las empresas modernas, ya que permite identificar y evaluar riesgos a los que están expuestas, especialmente aquellas que gestionan centros de datos (Datacenter). Estos entornos albergan información y activos críticos que sustentan los procesos operativos y estratégicos de la empresa. En un contexto donde las amenazas cibernéticas son cada vez más sofisticas y persistentes, resulta imperativo identificar de manera proactiva los riesgos asociados a la infraestructura tecnológica e implementar controles que garanticen la confidencialidad, integridad y disponibilidad de la información, en línea con lo establecido por la norma (ISO/IEC 27001, 2013).

Actualmente la empresa proveedora de servicios de Internet, incluye dentro de sus rutinas de auditorías de seguridad la evaluación del estado de su centro de datos mediante la técnica Caja Blanca. Este enfoque permite una revisión exhaustiva de los sistemas, ya que al administrador de red cuenta con pleno conocimiento e información interna y privilegios de acceso, segmentación de la red y políticas de seguridad Peltier (2016). A diferencia de otros enfoques, la auditoría de Caja Blanca brinda un análisis más profundo, enfocado en detectar vulnerabilidades estructurales y operativas que podrían comprometer la infraestructura tecnológica.

Los hallazgos obtenidos durante este proceso permitirán a la empresa proveedora de servicios de Internet (ISP) generar recomendaciones concretas, priorizar acciones correctivas y tomar decisiones estratégicas para endurecer la seguridad. Además, este proyecto contribuye al cumplimiento de normativas y buenas prácticas internacionales como las establecidas por el NIST (2020), el marco COBIT (ISACA, 2019) y la propia ISO/IEC 27002 (2022), las cuales exigen procesos continuos de evaluación, monitoreo y mejora.

Desde una perspectiva académica y profesional, este proyecto representa una oportunidad para aplicar conocimientos teóricos y prácticos en un entorno real, fomentando el desarrollo de competencias en ciberseguridad, análisis de riesgos y auditoría informática. Asimismo, aporta un valor tangible a la organización al ofrecer una visión clara del estado actual de su infraestructura tecnológica y las medidas necesarias para mitigar amenazas futuras, garantizando así la continuidad y resiliencia del negocio.

Definición del Problema

Antecedentes del Problema

La infraestructura de seguridad en los centros de datos es un factor importante para garantizar la integridad, confidencialidad y disponibilidad de los activos digitales que son importantes para la operación de empresas y gobiernos

Estas infraestructuras incluyen sistemas de perímetro, sistemas de prevención de intrusos (IPS), soluciones de anti-spam y antivirus, son elementos claves contra las amenazas cibernéticas. Sin embargo, estas tecnologías no están exentas de los ataques informáticos y pueden tener vulnerabilidades desconocidas o no parcheadas, errores en la configuración y limitaciones en su capacidad de detección y respuesta.

Cuando una infraestructura de seguridad en un centro de datos es comprometida, las consecuencias son severas y afectan directamente la operación como; interrupciones en el servicio, perdida de datos sensibles, impacto económico, amenazas a la continuidad del negocio etc.

Un ataque de ransonware en un centro de datos, por ejemplo, puede cifrar sistemas críticos para exigir un pago a cambio de restaurar el acceso. Un ejemplo de este tipo de ataque ocurrió el 7 de mayo del 2021 en Colonial Pipeline, quien opera el oleoducto de transferencia de combustible más grande de la costa este de los Estados Unidos. Este ataque genero retrasos en el suministro de combustible a lo largo de la costa este, lo que disparó el coste de la gasolina un 4 % (Pankov, 2021)

En Latinoamérica y el Caribe, la organización LACNIC recalca la importancia de seguridad en los centros de datos. Este enfoque es importante debido al papel esencial que desempeñan estas infraestructuras en la operación de servicios críticos, como la conectividad a

Internet y el almacenamiento de datos estratégicos. De acuerdo con LACNIC la región ha avanzado en la creación de Centros de Respuesta a Incidentes de Seguridad Informática (CSIRT), que son importantes para prevenir, detectar y mitigar incidentes de seguridad en centros de datos. Países como Bolivia y organizaciones en Belize han implementado CSIRTs con el apoyo de LACNIC, promoviendo prácticas robustas de ciberseguridad a través del proyecto AMPARO (Gianni, 2023)

En Colombia de acuerdo con ACIS (Asociación Colombiana de Informática, Sistemas y Tecnologías afines), el país sufrió un ciberataque masivo que afectó a más de 20 entidades públicas y 78 privadas, demostrando que el país está experimentando un aumento significativo en la cantidad y sofisticación de este tipo de amenazas *(ACIS, 2024*). Colombia ha desarrollado medidas para fortalecer la ciberseguridad en el país, como la creación de la Dirección de Ciberseguridad del Ministerio de las Tecnologías de la Información y las Comunicaciones (MinTIC), que tiene como objetivo formular la política de ciberseguridad del país, desarrollar y ejecutar programas y proyectos de ciberseguridad, promover la cooperación internacional y brindar asistencia técnica a las entidades públicas y privadas.

De acuerdo con el panorama de ciberseguridad para Colombia en este 2024, la entidad pionera en certificación y autenticación de identidad digital en Colombia Certicámara, emite una serie de recomendaciones de ciberseguridad para empresas públicas y privadas que se deben tener en cuenta:

Fortalecer sus sistemas de seguridad: Las organizaciones y empresas deben implementar sistemas de seguridad robustos que puedan detectar y responder a los ciberataques. Esto incluye la implementación de medidas de seguridad básicas, como firewalls, antivirus y protección

contra malware, así como la implementación de medidas más avanzadas, como la IA ACIS *(2024*)

Instruir a sus empleados en ciberseguridad: Tanto empresas, como empleados, deben estar capacitados en ciberseguridad para que puedan identificar y evitar los ciberataques. Esto incluye la capacitación en temas como la seguridad de la información, el phishing y el malware. *ACIS (2024*)

Formulación del Problema

¿Cómo se puede evaluar de manera eficiente el nivel de protección actual de la infraestructura de seguridad del centro de datos identificando vulnerabilidades y/o amenazas por medio de técnicas y herramientas de auditoria?

Objetivos

Objetivo General

Evaluar el nivel de protección de la infraestructura de seguridad, a través de técnicas y herramientas de auditoría, para la detección de vulnerabilidades y/o amenazas, en el centro de datos de una empresa proveedora de servicios de Internet (ISP) ubicado en la ciudad de Bogotá.

Objetivos Específicos

Establecer los activos de la infraestructura de seguridad, que serán elementos de análisis en las pruebas de auditoría.

Identificar las vulnerabilidades a la infraestructura de seguridad, mediante el uso de técnica caja blanca y siguiendo la metodología PTES.

Documentar los hallazgos obtenidos de las diferentes pruebas realizadas con la técnica Caja Blanca, en los activos de la infraestructura de seguridad de la empresa.

Proponer medidas y acciones que endurezcan la seguridad de la infraestructura del centro de datos de la compañía.

Marco Referencial

Marco Conceptual

Vulnerabilidad informática:

Una vulnerabilidad es un fallo en el sistema, servicio o configuración que puede ser explotado por una menaza para comprometer la seguridad del sistema Shostack (2014). Estas pueden producirse por errores en la configuración, fallos en el diseño, softwares desactualizados y errores humanos.

Amenaza informática:

Es cualquier posible peligro que pueda comprometer la seguridad de un sistema o de la información que este contiene.

Las amenazas son acciones que aprovechan una vulnerabilidad para comprometer la seguridad de un sistema de información. Estas tienen un efecto potencialmente negativo sobre los sistemas y pueden generarse desde distintos vectores de ataque como fraudes, robo, virus, sucesos físicos (incendios, inundaciones) o errores de configuración y decisiones institucionales (mal uso de contraseñas, falta de cifrado). Las amenazas pueden ser tanto internas como externas. (INCIBE, s.f.-b).

Hacker de sobrero Blanco:

Un Hacker de sombrero Blanco es un profesional que utiliza sus conocimientos para identificar vulnerabilidades en sistemas con el objetivo de corregirlas y prevenir posibles ciberataques.

Según. Kaspersky (2021), "Los hackers de sombrero blanco emplean sus habilidades para detectar fallos de seguridad y ayudar a las organizaciones a protegerse de hackers maliciosos.

Algunas empresas los contratan directamente para evaluar sus puntos débiles y fortalecer sus defensas".

Firewall:

Un Firewall es un software o dispositivo de seguridad diseñado para monitorear y controlar el tráfico de red, tanto entrante como saliente, mediante reglas de filtrado y análisis de datos dentro de una red informática. Su principal función es actuar como barrera de protección entre redes seguras y potenciales amenazas externas.

Como explica Fortinet (s. f.): "Los firewalls son soluciones de seguridad que protegen las redes bloqueando tráfico malicioso mediante reglas predefinidas. No solo previenen el acceso de malware, sino que también pueden restringir el acceso de usuarios internos a ciertos sitios o aplicaciones. Operan bajo el principio fundamental de autenticar e inspeccionar todo tráfico proveniente de zonas menos seguras antes de permitir su acceso a redes protegidas. Sin esta protección, los dispositivos de red quedan expuestos a ataques cibernéticos y vulnerables a intrusiones no autorizadas.

Centros de Datos:

Los centros de datos son instalaciones físicas o basados en la nube diseñados para alojar infraestructura tecnológica, aplicaciones y servicios digitales. Estas instalaciones concentran los recursos necesarios para el almacenamiento, procesamiento y distribución de datos.

Como señala AWS (s. f.) "Un centro de datos es una ubicación física que almacena máquinas de computación y sus equipos de hardware relacionados. Contiene la infraestructura computación que requieren los sistemas de TI, como servidores, unidades de almacenamiento de datos y equipos de red. Es la instalación física que almacena los datos digitales de cualquier empresa."

Marco Teórico

Este capítulo aborda los conceptos necesarios fundamentales como la seguridad de la información, seguridad perimetral, evaluación de vulnerabilidades, marcos normativos y técnicos que justifican su utilización en ambientes empresariales.

Seguridad de la Información y Ciberseguridad:

La seguridad de la información hace referencia a la protección de los activos informáticos frente a amenazas que comprometan su confidencialidad, integridad y disponibilidad (ISO/IEC, 2013). Por otro lado, la ciberseguridad amplía este concepto, incluyendo la defensa de redes, sistemas y servicios digitales ante ciberataques (Kim & Solomon, 2016). En ambos escenarios la detección temprana de vulnerabilidades es esencial para la prevención de incidentes y la continuidad operativa de los servicios tecnológicos.

Infraestructura de Seguridad en Centro de Datos:

El centro de datos hospeda la infraestructura tecnológica crítica de una organización. Su seguridad está compuesta con múltiples capas, incluyendo firewalls, IDS/IPS, segmentación de red, antivirus, control de accesos y monitoreo activo (Peltier, 2016). La gestión eficaz de estos elementos requiere una configuración adecuada y la supervisión constante de posibles fallos o brechas de seguridad que puedan ser explotadas por actores maliciosos.

Evaluación de Vulnerabilidades:

La evaluación de vulnerabilidades es un procedimiento técnico y metodológico enfocado en la identificación, clasificación y remediación de fallos de seguridad en redes, sistemas operativos, aplicaciones y dispositivos conectados (Scarfone & Mell, 2007). Este análisis se basa en escaneos automatizados y revisiones manuales, permitiendo detectar configuraciones erróneas, servicios expuestos, software desactualizado o códigos inseguros.

Normas y Estándares Aplicables:

La aplicación de estándares reconocidos aporta estructura, objetividad y credibilidad al proceso de evaluación. Entre los principales se destacan:

- ISO/IEC 27001:2013, que establece los requisitos para establecer, implementar y mantener un Sistema de Gestión de Seguridad de la Información (SGSI) (ISO/IEC, 2013).
- ISO/IEC 27002:2022, que proporciona controles y buenas prácticas para la protección de los activos de información (ISO/IEC, 2022).
- NIST SP 800-115, que ofrece guías detalladas para la planificación y ejecución de pruebas técnicas de seguridad (NIST, 2008).
- OWASP Top Ten, que categoriza las vulnerabilidades más frecuentes en aplicaciones web y propone controles de mitigación (OWASP, 2021).
- CIS Controls, conjunto de buenas prácticas para defender sistemas ante amenazas comunes y priorizar recursos de seguridad (Center for Internet Security, 2021).

Herramientas para el Análisis de Vulnerabilidades:

- Múltiples herramientas especializadas permiten realizar escaneos y evaluaciones técnicas de seguridad:
- OpenVAS: escáner de vulnerabilidades de código abierto que analiza servicios, puertos y configuraciones inseguras.
- Nessus: plataforma comercial ampliamente utilizada para detección y gestión de vulnerabilidades.
- Nmap: utilidad para mapeo de redes y detección de puertos y servicios activos (Lyon, 2009).

- Metasploit Framework: entorno para pruebas de penetración que permite validar la explotación de vulnerabilidades detectadas (Maynor, 2011).

Metodologías de Evaluación:

Las metodologías estandarizadas permiten estructurar el proceso de evaluación de vulnerabilidades y pruebas de seguridad. Entre las más comunes se encuentran:

- OSSTMM (Open Source Security Testing Methodology Manual): metodología abierta para pruebas de seguridad operativa.
- PTES (Penetration Testing Execution Standard): define fases y procedimientos para pruebas de penetración éticas.
- Metodología OWASP: orientada específicamente a pruebas de seguridad en aplicaciones web

Estas metodologías facilitan la documentación de hallazgos, la trazabilidad de acciones y la evaluación de impacto.

Gestión del Riesgo:

La evaluación de vulnerabilidades está directamente relacionada con la gestión del riesgo informático. Este proceso permite identificar amenazas, valorar su probabilidad e impacto, y priorizar acciones de mitigación (ISO/IEC, 2013). Una correcta gestión del riesgo contribuye a la toma de decisiones acorde con criterios técnicos y estratégicos, alineados con los objetivos del negocio.

Diseño de Metodología

El presente trabajo se desarrollará bajo un enfoque cuantitativo, haciendo uso de técnicas de recolección de datos numéricos con el fin de identificar y evaluar vulnerabilidades en la infraestructura de seguridad del centro de datos de la empresa proveedora de servicios de Internet (ISP). La metodología cuantitativa permite evaluar las vulnerabilidades y los riesgos asociados mediante las métricas, como por ejemplo los CVSS (Common Vulnerability Scoring System), que proporcionan datos medibles y comparables.

Los CVSS se considera una herramienta clave para la medición objetiva de vulnerabilidades. De acuerdo con el documento oficial de especificaciones de FIRST (2019), CVSS permite evaluar el impacto de las vulnerabilidades en función de factores intrínsecos y específicos del entorno, brindando un marco sólido para la priorización de esfuerzos de mitigación.

El proyecto se desarrollará implementando la metodología PTES (Penetration Testing Execution Standard) adaptada para un análisis de vulnerabilidades con la técnica Caja Blanca, excluyendo las fases de explotación y post-explotación. También se integrarán principios de NIST SP 800-115 y OSSTMM garantizando su ejecución de acuerdo con las buenas prácticas.

La metodología PTES, es la más adecuada para la ejecución del proyecto debido a:

- Estructura solida: PTES brinda un marco detallado que involucra todas las etapas necesarias para la ejecución de pruebas de seguridad, desde la definición del alcance hasta la documentación de los hallazgos.
- Flexibilidad: PTES permite adaptar sus etapas de acuerdo con las necesidades del proyecto, lo cual es importante para un análisis de vulnerabilidades cuando se realizan en entornos de producción.

- Enfoque Integral: PTES integra las buenas prácticas metodológicas reconocidas como NIST SP 800-115 y OSSTMM, garantizando la cobertura de todos los aspectos críticos de seguridad.
- Priorización basada en riesgos: Facilita la identificación y clasificación de las vulnerabilidades mediante técnicas estandarizadas como CVSS, lo que permite la toma de decisiones con fundamento.
- Relevancia académica y practica: PTES es una metodología ampliamente reconocida en el campo de la seguridad informática, la ejecución de esta metodología otorga validez y rigor al proyecto.

Metodología PTES

El proyecto de grado aplicado Análisis de vulnerabilidades en infraestructura de seguridad y datos de la compañía proveedora de servicios de Internet (ISP), se desarrollará adoptando la metodología PTES de acuerdo con los requerimientos actuales de la compañía.

La metodología PTES, cuenta con 7 fases que se describirán a continuación y de las cuales adoptaremos las fases que más se ajusten al proyecto.

Fase Interacciones Previas

El objetivo de esta sección del PTES es presentar y explicar las herramientas y técnicas disponibles que ayudan en un paso de pre-compromiso de una prueba de penetración. (Pre-engagement - the Penetration Testing Execution Standard, s. f.).

Se realizaron reuniones previas con la dirección de ingeniería de la compañía proveedora de servicios de Internet (ISP), en la que se presentó la propuesta de proyecto aplicado, sus objetivos, alcances, posibles resultados e impacto que este proyecto genera a la compañía. Durante esta fase se definió el alcance del proyecto estableciendo que es necesario ajustar la metodología de penetración propuesta y excluir las fases de Explotación y Pos Explotación, tendiendo en cuenta que el proyecto se realizará sobre infraestructura en producción. Se define que la infraestructura a analizar es la de seguridad incluyendo equipos de comunicaciones como router y switch. Se define la técnica Caja Blanca para el proyecto. En común acuerdo con las partes interesadas, se firman documentos de confidencialidad y autorizaciones para avalar el proyecto y seguir con las fases siguientes.

Recopilación de la Información

El objetivo de esta fase es crear un proceso especifico y detallado para poder realizar el reconocimiento del objetivo a analizar y recopilar la mayor información posible sobre el

objetivo. Para ello se implementa un modelo de tres niveles (Pasiva, semi pasiva y activa) complementado con la metodología

Open Source Intelligence (OSINT) en cada uno de sus tres niveles.

A continuación, se describen los niveles de recopilación de la información:

Nivel 1: Recopilación Pasivo (Cumplimiento Normativo)

Este nivel no se interactúa directamente con los equipos objetivo. Se basa en el uso de herramientas automatizadas, revisión de manuales y configuraciones proporcionados por la empresa proveedora de servicios de Internet (ISP). Además, se consulta en bases de datos públicas para buscar vulnerabilidades conocidas CVE asociados a los dispositivos y software utilizados. Este nivel es adecuado para cumplir con los requisitos normativos como PCI, FISMA o HIPAA, ya que brinda el mínimo necesario para demostrar que se ha realizado un esfuerzo de recopilación de información.

Nivel 2 Recopilación Semi-pasivo (Mejores Prácticas)

Este nivel, se realizan consultas indirectas sin generar actividad notable en los equipos, las consultas se realizan en plataformas como shodan, whois o Censys con el fin de obtener información sobre equipos expuestos en internet. Este nivel combina el uso de herramientas automatizadas con análisis manual, logrando una compresión más profunda del entorno, como ubicaciones físicas, relaciones comerciales y estructura organizacional

Nivel 3 Recopilación Activa. (Ataques avanzados o Red Team)

En esta fase se interactúa directamente con los equipos para recopilar información de manera no intrusiva y controlada. Las tareas en esta fase comprenden en realizar escaneos ligeros con nmap ejecutando sus módulos no intrusivos, uso de SNMPwalk para la extracción de configuraciones en los dispositivos.

Modelado de Amenazas

En esta fase se analiza los datos obtenidos de la fase de recopilación de información determinando los tipos de amenazas y actores que podrían comprometer los activos previamente identificados siguiendo los siguientes 4 pasos:

1. Reunir documentación Pertinente.

Revisar la documentación obtenida en el OSINT pasivo; documentación, fichas técnicas, configuraciones, topologías y complementarla con la información del OSINT Semi Pasivo.

2. Identificar y categorizar los activos primarios y secundarios:

Se creará un inventario detallado de los activos primarios como servidores firewall y dispositivos firewall y activos secundarios como switches y routers.

3. Identificar y categorizar las comunidades de amenazas y amenazas.

En esta fase se identifica la comunidad de amenazas como las amenazas externas como, por ejemplo: Hakers, malware automatizado. Amenazas Internas como: Empleados maliciosos, errores humanos de configuración y ambientales como interrupciones en el servicio de energía.

4. Mapear comunidades de amenazas contra los activos primarios y secundarios.

En este paso se relaciona las comunidades de amenazas y las amenazas identificadas en los activos críticos priorizando los escenarios de mayor riesgo. Desarrollar una matriz de riesgos con los activos primarios y secundarios.

Análisis de Vulnerabilidades

En esta fase, se ejecutan pruebas estructuradas con interacción directa y controlada sobre los dispositivos identificados, con el propósito de detectar y clasificar vulnerabilidades presentes en la infraestructura de red. Este proceso se enfoca principalmente en las tres primeras capas del

modelo OSI (física, enlace de datos y red), donde operan dispositivos críticos como switches, routers y firewalls.

El análisis de vulnerabilidades empleará herramientas automatizadas, como OpenVAS o Nessus, para escanear configuraciones, servicios activos y posibles fallos de seguridad. Estas herramientas permiten identificar vulnerabilidades conocidas (basadas en CVE) y calcular su criticidad según métricas estandarizadas como CVSS, optimizando tiempo y esfuerzo del análisis.

Adicionalmente, se realizarán verificaciones manuales para validar los hallazgos más relevantes y descartar falsos positivos, asegurando la precisión del análisis. Este enfoque, alineado con el modelo PTES, permite priorizar las vulnerabilidades detectadas según su impacto potencial y su probabilidad de explotación, proporcionando una base sólida para las recomendaciones de mitigación en etapas posteriores.

Explotación

En esta fase, se ejecuta la explotación de las vulnerabilidades detectadas, con el objetivo de validar su impacto y demostrar la posibilidad de acceder a los dispositivos o sistemas evitando las medidas de seguridad configuradas. Sin embargo, dado que la infraestructura evaluada se encuentra en un entorno de producción, y conforme a los acuerdos establecidos en la fase de interacciones previas, no se llevará a cabo la explotación activa de las vulnerabilidades.

Post-Explotación

La fase de post-explotación, se centra en establecer persistencia en los sistemas comprometidos y evaluar la capacidad para mantener el control de los dispositivos analizados. Esta fase permite explorar cómo el acceso inicial podría utilizarse para escalar privilegios y comprometer otros activos dentro de la red, según su criticidad y función.

Sin embargo, debido a que la infraestructura evaluada se encuentra en un entorno de producción, esta fase no será realizada en el presente proyecto, en cumplimiento de los acuerdos establecidos en la fase de interacciones previas.

Reporte

Al finalizar el análisis de vulnerabilidades en los dispositivos de seguridad y red, se elaborará un informe ejecutivo diseñado para comunicar de manera clara y efectiva los objetivos, métodos y resultados del pentesting realizado en la infraestructura de la empresa proveedora de servicios de Internet (ISP). Este informe estará estructurado para ser comprensible tanto para el personal técnico y personal responsable de la toma de decisiones y otras partes interesadas, facilitando la comprensión de los hallazgos y su impacto.

Además, se desarrollará un informe técnico detallado que documentará de forma exhaustiva el proceso ejecutado, desde las fases iniciales de recolección de datos hasta el análisis de vulnerabilidades. Este informe incluirá una descripción de las herramientas empleadas en cada etapa, los hallazgos obtenidos, y un plan de acción propuesto para fortalecer la seguridad de la infraestructura en la empresa proveedora de servicios de Internet (ISP), asegurando que los activos críticos estén protegidos frente a posibles amenazas.

Desarrollo de Los Objetivos

Establecer los Activos de la Infraestructura de Seguridad

El desarrollo de este objetivo se basa en la metodología PTES (Penetration Testing Execution Standard). En esta fase se determinó el alcance del proyecto, incluyendo la identificación y listado de activos críticos que serán analizados. Estos activos comprenden los equipos de seguridad y de red que forman parte de la infraestructura del centro de datos de la empresa proveedora de servicios de Internet (ISP), tales como firewall, servidores VPN, switches y routers. La identificación de estos activos es importante para garantizar que el análisis de vulnerabilidad sea exhaustivo y se enfoque en los componentes más críticos de la empresa.

Fase Interacciones Previas

De acuerdo con la metodología PTES, el proyecto inició con la fase de interacciones previas. En esta fase se llevaron a cabo reuniones con el departamento de ingeniería de la empresa proveedora de servicios de Internet (ISP), en las cuales se presentó el alcance del proyecto, objetivos, la importancia de la evaluación de seguridad y el impacto esperado en la infraestructura de la empresa. Durante estas reuniones se definió que el análisis se centraría en la infraestructura de seguridad del centro de datos, con el fin de identificar vulnerabilidades en los equipos de red y seguridad.

En esta fase se acordó que el modelo de pruebas de penetración (PenTesting) a ejecutar sería el de Caja Blanca (White-Box), lo que implica que el evaluador tendrá acceso completo a la información de los dispositivos, incluyendo configuraciones, topologías de red y documentación técnica. Este enfoque permite un análisis a profundidad y detallado, y que el evaluador cuenta con conocimiento completo de la infraestructura, lo que facilita la identificación de vulnerabilidades tanto a nivel de configuración como de diseño.

Además, se establecieron acuerdos de confidencialidad y autorizaciones necesarias para garantizar que el proyecto se realice de manera controlada y segura, sin afectar las operaciones críticas de la empresa. Dichos acuerdos incluyen la exclusión de las fases de explotación y post-explotación, ya que el análisis se realizará en un entorno de producción, lo que implica que no se ejecutarán acciones que puedan comprometer la disponibilidad o integridad de la red.

Recopilación de la Información

Recopilación Pasiva de la Información

En esta fase se recopila la mayor cantidad de información posible sobre los activos de la infraestructura de seguridad que será objeto de análisis. Dado que el proyecto se basa en la técnica Caja Blanca, se obtiene acceso autorizado y completo a la información detallada de los equipos, subredes y direcciones IPv4/Ipv6 que componen la infraestructura de seguridad de la empresa proveedora de servicios de Internet (ISP), de acuerdo con lo definido previamente en la fase de interacciones previas.

Con la información proporcionada por la empresa proveedora de servicios de Internet (ISP), se accede a la documentación técnica relevante como listas de dispositivos, diagramas y configuraciones específicas. Con la información recopilada, se elabora un formato de levantamiento de activos que sirve como base para el análisis de vulnerabilidades, garantizando que todos los componentes críticos sean identificados, categorizados y priorizados adecuadamente.

Tabla 1

Levantamiento de la Información

	Recolección de Información		
Nombre del Proyecto	Análisis de Vulnerabilidades en Infraestructura de Seguridad.	Fecha de Solicitud	29/01/2025
Empresa	Empresa Proveedora de Servicios de Internet (ISP)	Fecha Estimada de Ejecución.	5/02/2025
Responsable del Proyecto	Carlos Holmes Fernández Rivera		
Tipo de Pentest	Caja Blanca		
Alcance del Pentest	Análisis de vulnerabilidades en dispositivos de seguridad y red.		
Descripción del Alcance	Análisis de vulnerabilidad en equipos de seguridad y red, empleando la técnica Caja Blanca.		
Entorno	Infraestructura de Seguridad en Producción		

Direcciones IP y Rango de Subredes a Evaluar.

Rango/Subred	CIDR	Descripción del Activo
200.xx.xx.0/24	200.xx.xx.x28/26	Segmento Público WAN
200.xx.xx.0/24	200.xx.xx.x52/30	Segmento Público WAN
200.xx.xx.0/24	200.xx.xx.x28/26	Segmento Público WAN
192.168.xxx.xx/24	192.168.xx.xx/24	Segmento LAN Mangmen
192.168.xxx.xx/24	192.168.xx.xx/24	Segmento LAN administración

Nota. Recolección de información básica del proyecto.

Tabla 2

Activos para Evaluar

Activos Para Evaluar				
ítem	Nombre del activo	Tipo de Activo	IP	Comentarios
1	Firewall Principal	Firewall	192.xxx.xxx.x26	No reiniciar durante las pruebas
2	Firewall Infraestructura	Firewall	192.xxx.xxx.x01	No reiniciar durante las pruebas
3	Firewall Corporativos	Firewall	200.xx.xxx.x30	No reiniciar durante las pruebas
4	Servidor VPN Mikrotik	Router Mikrotik servidor	200.xx.xxx.x54	No reiniciar durante las pruebas.
5	Router Principal	Router	200.xx. xx.x29	No reiniciar durante las pruebas.
6	Switch distribución	Switch	192.xx.xx.x54	No reiniciar durante las pruebas.
7	Switch de acceso	Switch	192.xxx.xx.x00	No reiniciar durante las pruebas.
8	Switch de acceso	Switch	192.xxx.xx.x01	No reiniciar durante las pruebas.
9	Switch de acceso	Switch	192.xxx.xx.x02	No reiniciar durante las pruebas.
10	Switch de acceso	Switch	192.xxx.xx.x04	No reiniciar durante las pruebas.

Activos Para Evaluar				
ítem	Nombre del activo	Tipo de Activo	IP	Comentarios
11	Switch de acceso	Switch	192.xxx.xx.x03	No reiniciar durante las pruebas.
12	Switch de acceso	Switch	192.xxx.xx.x90	No reiniciar durante las pruebas.
13	Servidor OpenVpn/Ipsec	Firewall	200.xx.xxx.x30	No reiniciar durante las pruebas.
14	Servidor VPN IPsec	Firewall	200.xx.xxx.x30	No reiniciar durante las pruebas.

Nota. Recolección de información activos a evaluar.

Clasificación de Activos

Para el análisis de vulnerabilidades en la infraestructura de seguridad del centro de datos, se implementa un sistema de clasificación de activos basado en su criticidad en caso de compromiso o falla, agrupándolos en dos categorías principales: activos en Primarios y Secundarios.

Activos Primarios. Aquellos dispositivos que son críticos en la operación del centro de datos y cuya falla o intermitencia genera un impacto significativo en la continuidad del negocio, la seguridad de la información o disponibilidad de los servicios. Estos activos se encuentran mayormente expuestos a amenazas exteriores al tener configuraciones con direcciones ipv4/ipv6 públicas por lo que se requiere de un mayor nivel de protección.

Activos Secundarios. Aquellos dispositivos que son importantes, pero no representan un impacto critico en la operación del centro de datos si se ven comprometidos. La falla o

compromiso puede afectar la eficiencia o seguridad, pero no necesariamente la paralización de la operación.

Modelo de priorización

Se implementa un modelo de priorización para cada activo basado en tres criterios fundamentales:

Impacto en la continuidad del negocio

Tabla 3

Criticidad en la Continuidad del Negocio

ALTA	Activos que pueden paralizar o causar interrupción significativa al ser comprometidos o fallas en el equipo
MEDIA	Activos que pueden disminuir o ralentizar la eficiencia o seguridad, sin causar parálisis de la operación.
BAJA	Activos cuya falla o compromiso generan un impacto mínimo en el centro de datos.

Nota. Esta tabla define la criticidad del negocio.

Exposición de Amenazas:

Tabla 4

Criticidad Exposición de Amenazas

ALTA	Activos que se encuentran expuestos a amenazas externas o internas debido a su ubicación en la red o su función.
MEDIA	Activos que se encuentran menos expuestos a amenazas directas, sin embargo, podrían ser objetivos de ataques.
BAJA	Activos que se encuentran menos expuestos a amenazas de acuerdo con su ubicación en la red o su función.

Nota. Esta tabla define la criticidad de amenazas.

Criticidad de la Información:

Tabla 5

Criticidad en la Información

ALTA	Activos que se encuentran expuestos a amenazas externas o internas debido a su ubicación en la red o su función.
MEDIA	Activos que se encuentran menos expuestos a amenazas directas, sin embargo, podrían ser objetivos de ataques.
BAJA	Activos que se encuentran menos expuestos a amenazas de acuerdo con su ubicación en la red o su función.

Nota. Esta tabla define la criticidad de la información.

Tabla 6

Clasificación de Activo Según Criticidad

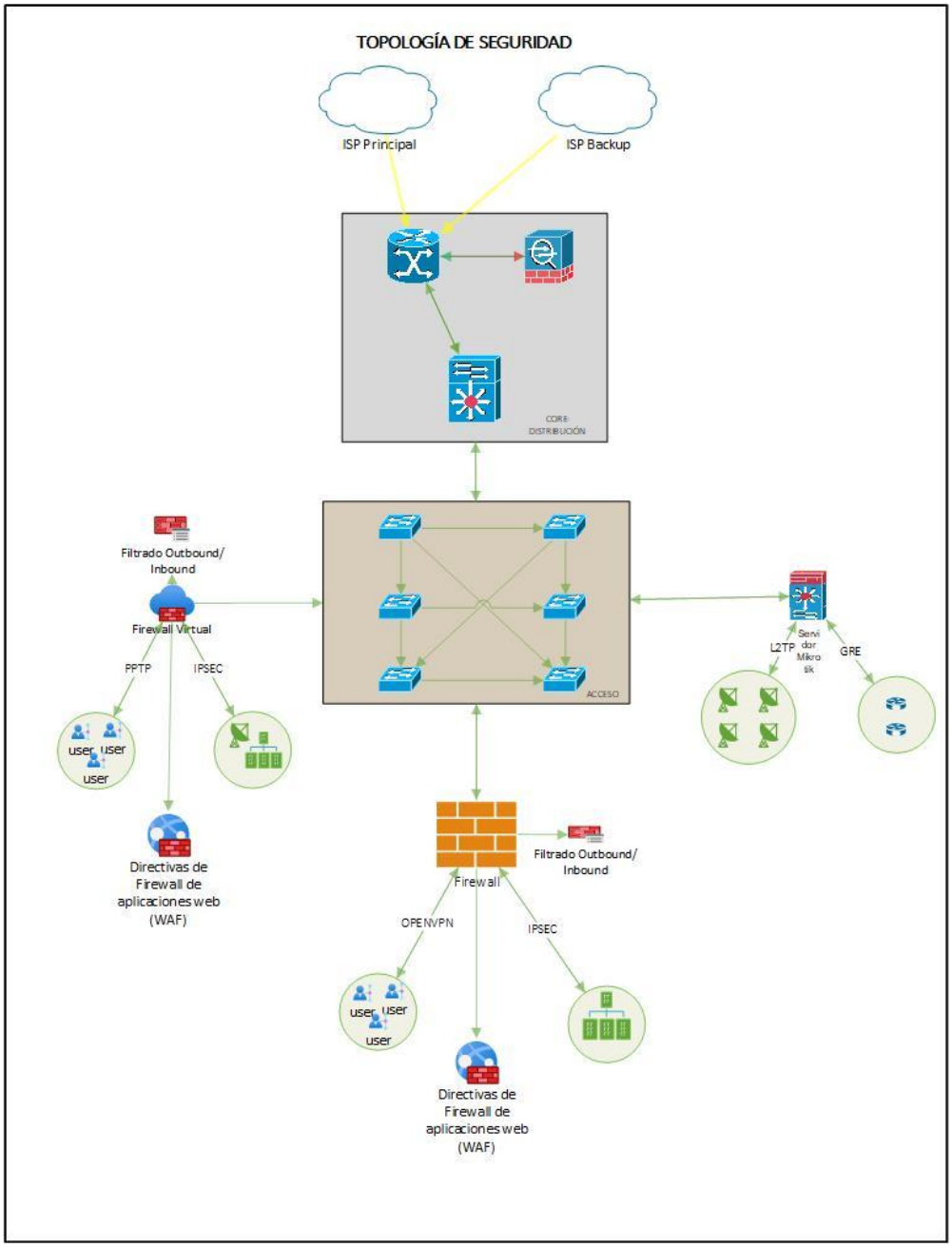
Inventario de Activos								
ID	Cantidad	Descripción	Tipo	Ubicación	Continuidad del Negocio	Exposición a Amenazas	Criticidad de la Información	Prioridad
1	1	Router Core	Hardware	Datacenter	Alta	Alta	Alta	Alta
2	1	Firewall Core	Hardware	Datacenter	Alta	Alta	Alta	Alta
3	1	Firewall Corporativos	Hardware	Datacenter	Alta	Alta	Alta	Alta
4	1	Swirch de Distribución	Hardware	Datacenter	Alta	Media	Alta	Alta
5	1	Firewall Infrastructura	Software	Datacenter	Media	Alta	Media	Media
6	1	Servidor VPN Mikrotik	Hardware	Datacenter	Media	Alta	Media	Media
7	1	Servidor OpenVpn/Ipsec	Software	Datacenter	Media	Alta	Media	Media
8	1	Servidor VPN Ipsec	Software	Datacenter	Media	Alta	Media	Media
9	6	Switch de Acceso	Hardware	Datacenter	Baja	Baja	Baja	Baja

Nota. Esta tabla clasifica los activos de acuerdo con su criticidad en la infraestructura.

Con la información obtenida en la fase Pasiva, se elabora un diagrama lógico de la infraestructura de seguridad.

Figura 1

Topología de Infraestructura de Seguridad



Nota. Elaboración propia en Visio diagrama topológico de la infraestructura.

Recopilación Semi-Pasivo de la Información

Esta fase, se realizará un análisis detallado, combinando técnicas manuales y automatizadas, para procesar y complementar la información obtenida en la fase de recopilación pasiva. Además, se llevará a cabo una búsqueda automatizada de dispositivos expuestos en internet, con el fin de identificar posibles vectores de ataques.

Mediante el análisis de los datos obtenidos en la fase 1, se determinó que los activos con criticidad alta se encuentran directamente expuesto a Internet. En particular el equipo router de Core, cuyas interfaces de borde están configuradas con direcciones IPv4 públicas y IPv6 Unicast Global, los que los convierte en puntos críticos para la seguridad de la infraestructura.

Para identificar activos expuestos en Internet, se desarrollaron scripts en Python que automatizan la recopilación de información mediante fuentes como Shodan. Esta estrategia permite una detección más precisa y eficiente, reduciendo la intervención manual y el margen de error.

Herramientas y Bibliotecas Utilizadas

Python: Es un lenguaje de programación de alto nivel interpretado, se caracteriza por su sintaxis clara y legible. Fue creado por Guido van Rossum y lanzado por primera vez en 1991 (Python Software Foundation, 2023). Python destaca por su extensa biblioteca y una comunidad activa, Python es ampliamente utilizado en desarrollo web, análisis de datos, inteligencia artificial, automatización de tareas, ciberseguridad.

Shodan: Motor de búsqueda especializado en recopilación de información sobre la infraestructura de red y servicios expuesto públicamente como; servidores, cámaras web, routers y otros dispositivos de IoT (Shodan, 2023). Esta herramienta permite identificar dispositivos

conectados a Internet y sus configuraciones, lo que es útil para el análisis de vulnerabilidades y la evaluación de riesgos en entornos de red.

Conda: Gestor de paquetes y entornos de código abierto que permite a los usuarios instalar, ejecutar y actualizar paquetes y sus dependencias en múltiples plataformas (Anaconda, Inc., 2023). En el ámbito de ciberseguridad y análisis de vulnerabilidades, Conda es especialmente útil porque permite la creación de entonos aislados para ejecutar herramientas de seguridad y análisis sin interferir con otros sistemas. Esto garantiza que los profesionales de ciberseguridad probar y evaluar herramientas de manera segura y eficiente.

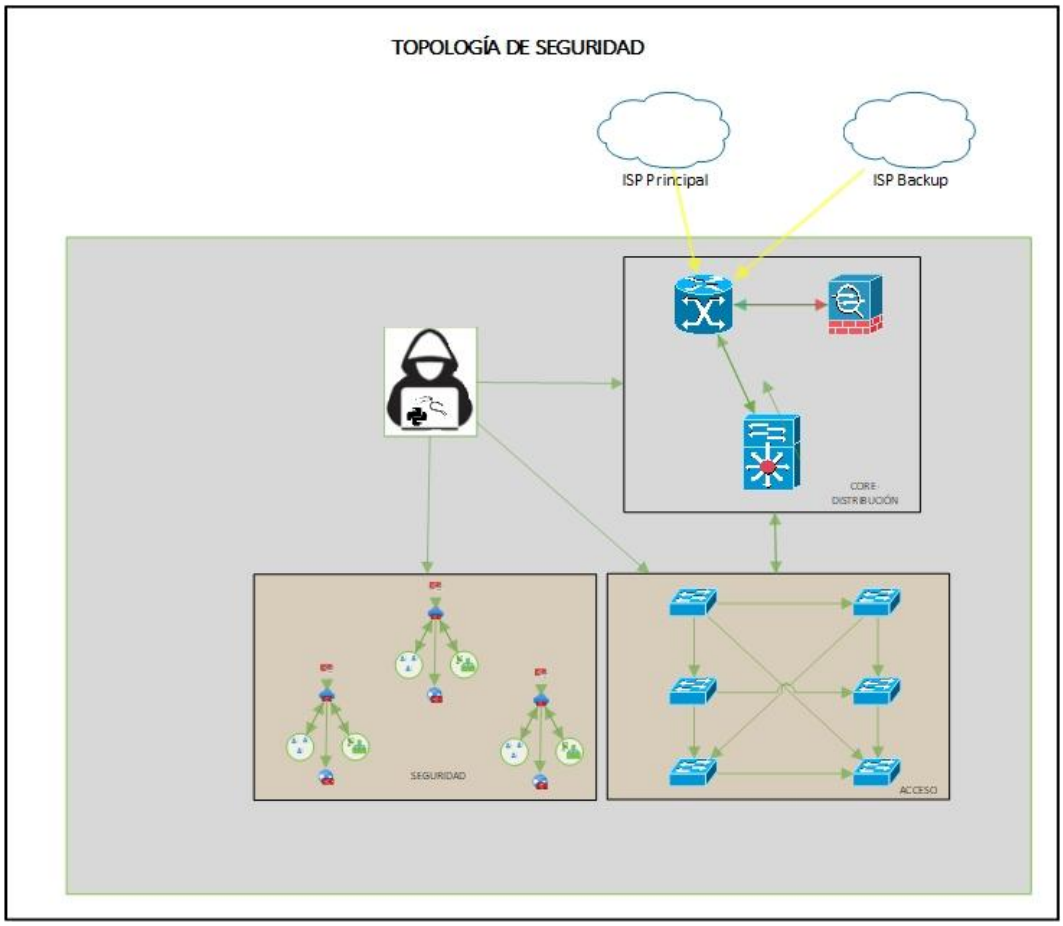
Kali Linux: Es una distribución de Linux Open-Source basada en Debian, diseñada específicamente para realizar pruebas de penetración avanzadas y auditorias de seguridad (¿What is Kali Linux? | Kali Linux Documentation, s.f.). Se ejecuta en múltiples plataformas y es ampliamente utilizado por profesionales de ciberseguridad debido a su extenso repertorio de herramientas, configuraciones y scripts los cuales incluyen modificaciones específicas para la industria. Estas características lo convierten en una opción ideal para tareas como el análisis forense, ingeniería inversa, análisis de vulnerabilidades, recopilación de la información y pruebas de seguridad en redes y sistemas.

Preparación del Entorno

Para la ejecución de esta fase, se ha configurado una máquina virtual dentro del segmento de administración de la infraestructura de seguridad, garantizando visibilidad sobre todos los dispositivos de red y seguridad. La máquina cuenta con Kali Linux actualizado y preconfigurado con bibliotecas y herramientas necesarias para las siguientes fases de análisis.

Figura 2

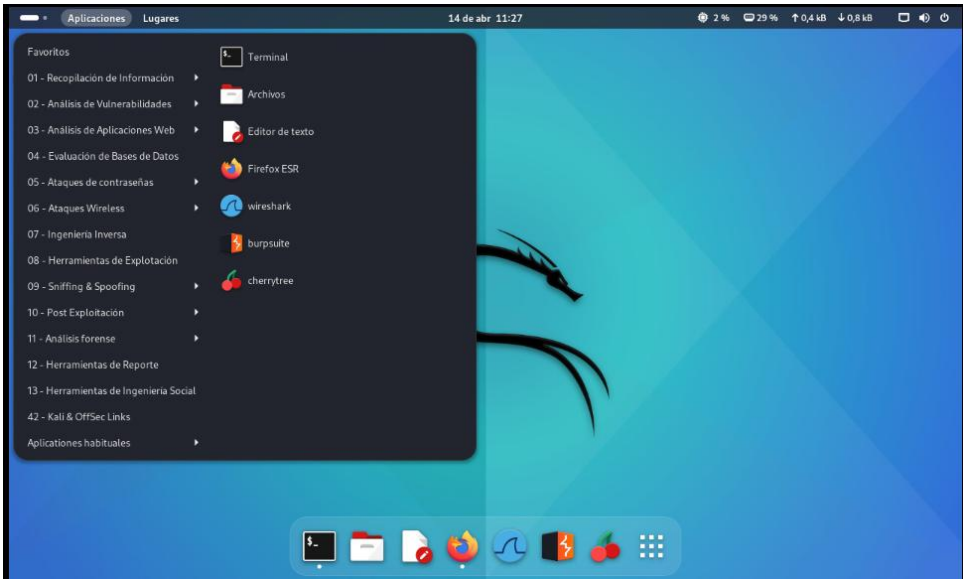
Preparación del Entorno para el Análisis de Vulnerabilidades.



Nota. Elaboración propia en Visio diagrama topológico de la infraestructura.

Figura 3

Entorno Virtual Kali Linux.



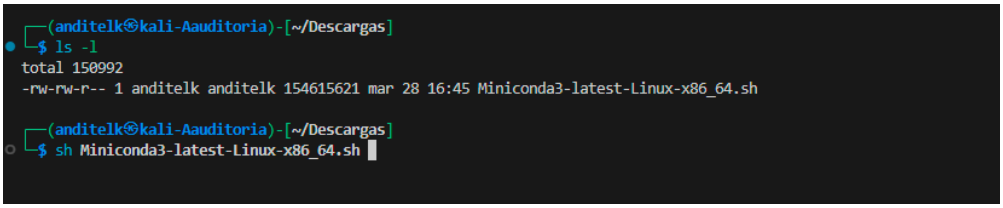
Nota. Pantallazo entorno de escritorio Kali Linux.

La creación de entornos virtuales permite aislar el entorno de ejecución de un proyecto Python del sistema operativo principal y de otros proyectos de desarrollo. Esta practica resulta esencial en proyectos de ciberseguridad y automatización, ya que asegura que las versiónes de las bibliotecas y dependencias utilizadas no se interfieran entre si, ni con herramientas instalas en el sistema.

La siguiente ilustración muestra el proceso de creación de un entorno virtual para la fase de recopilacion semi-pasiva de la información implementado con Conda.

Figura 4

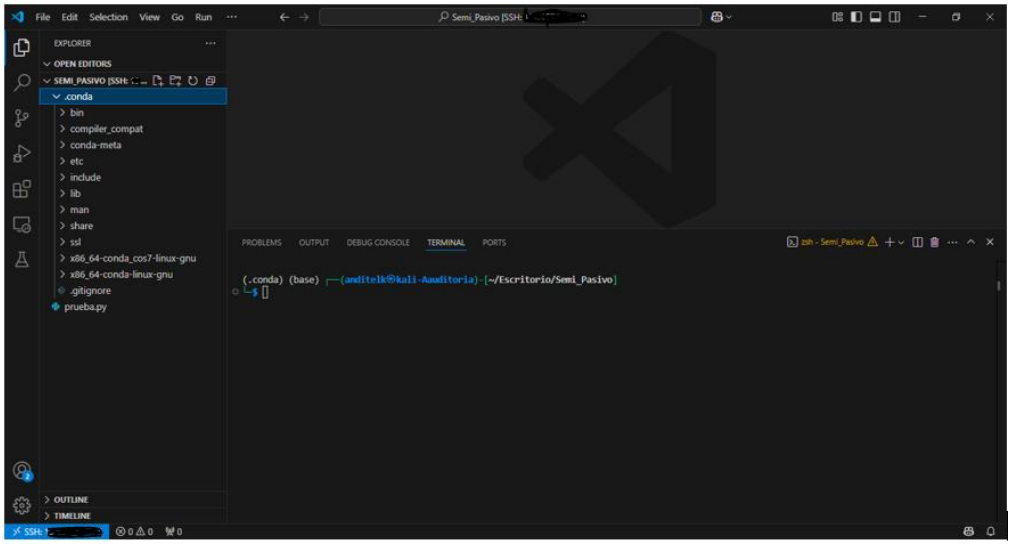
Descarga del Paquete Miniconda.



Nota. Ejecución del script de instalación de Miniconda,

Figura 5

Preparación del Entorno Virtual en Python.



Nota. Preparación del entorno de trabajo en Visual Studio Code.

Instalación de Shodan en el ambiente virtual.

Shodan es una herramieta especializada en la recolección de información sobre dispositivos conectados a Internet. Shodan a diferencia de los navegadores tradicionales como Google, Firefox que indexan sitios web, shodan indexa dispositivos de red como router ,

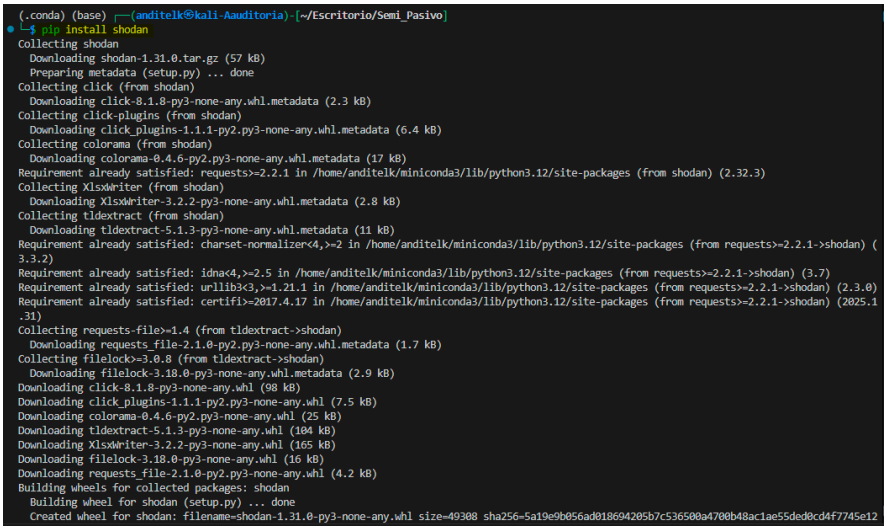
camarás, firewall, servidores VPN entre otros, revelando, servicios, puertos abiertos y enmuchos casos versiones de software y configuraciones visibles externamente.

Dentro del marco del presente proyecto, Shodan será implementado durante la fase de recopilacipon semi-pasiva de la información, permitiendo detectar activos de la infraestructura de seguridad de la empresa proveedora de servicios de Internet (ISP), que se encuentren expuestos a internet, ya sea por medio de sus interfaces públicas o servicios visibles a través de direcciones IPv4 y IPv6.

La siguiente ilustraciónes muestra el proceso de instalación de Shodan.

Figura 6

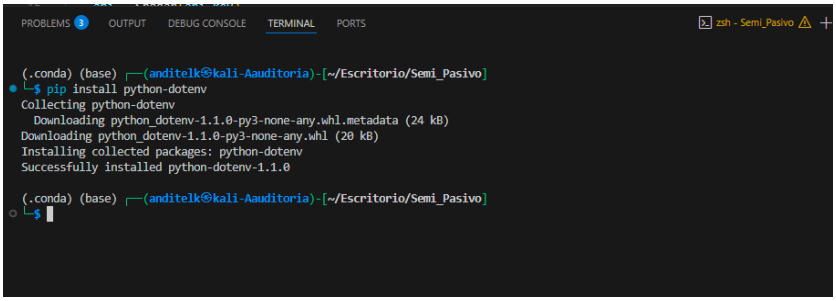
Instalación de la Herramienta Shodan.



Nota. Instalación de programa Shodan.

Figura 7

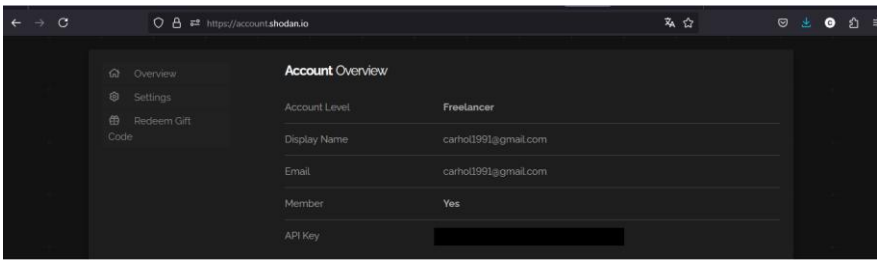
Instalación del Paquete Dotenv.



Nota. Instalación de programa Dotenv con pip.

Figura 8

Interfaz Web de la Herramienta Shodan.



Nota. Captura de pantalla entorno de trabajo Shodan.

Desarrollo del Script Python.

El desarrollo del script en lenguaje Python, permite integrar de forma estructurada consultas hacía la API de Shodan, centralizando el analisis de exposición de activos criticos de la infraestructura de seguridad. La implementanción del script permite estandarizar procesos de consulta, minimizar el error humano en busquedas manuales, agilizar el analis de segmentos de red establecidos en la Fase 1 y almacenar resultados para la correlación furuta con herramientas de analsis de vulnerabilidades.

Importanción de librerias.

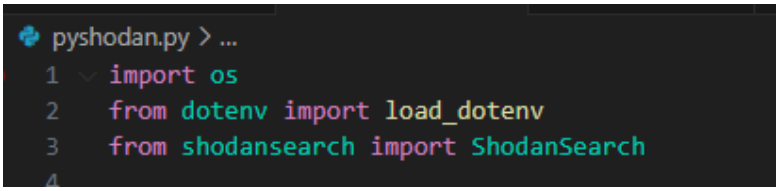
Las librerias cumplen una función especifica clave para la automatización segura y eficiente.

- Os: Es una librería estándar de Python que permite interactuar con el sistema operativo. Esta librería se usa para acceder a las variables de entorno, donde se almacena de manera segura la clave API de Shodan (SHODAN_API_KEY). De esta forma, se evita exponer la clave directamente en el código fuente.
- Dotenv: Permite cargar automáticamente variables definidas en un archivo .env al entorno de ejecución del script. Este archivo almacena configuraciones sensibles, como claves y tokens de autenticación, que deben mantenerse fuera del control de versiones y protegidas del acceso no autorizado.
- El uso de .env permite que el script pueda ejecutarse en distintos entornos sin necesidad de modificar su código, favoreciendo su portabilidad y seguridad.
- Es un módulo personalizado desarrollado como parte del proyecto. Define una clase ShodanSearch, que encapsula toda la lógica relacionada con la conexión a la API de Shodan, la construcción de las consultas, el manejo de respuestas y el procesamiento de resultados.

Este enfoque modular y reutilizable permite mantener el script principal más limpio y facilita futuras ampliaciones, como filtrado de resultados, integración con otras herramientas, o generación automatizada de reportes.

Figura 9

Librerías Necesarias para Construcción del Script.



Nota. Captura de pantalla librerías necesarias para el script.

El siguiente fracmento de código desarrollado en python, tiene como objetivo inicializar el entorno para realizar busquedas en shodan. Shodan, plataforma que permite identificar dispositivos conectados a Internet. La automatización forma parte del proceso semi-pasivo de la información, ya que la busqueda no genera trafico directo a los activos objetivos, si no que realiza la consulta en los datos recopilados por Shodan.

Tabla 7

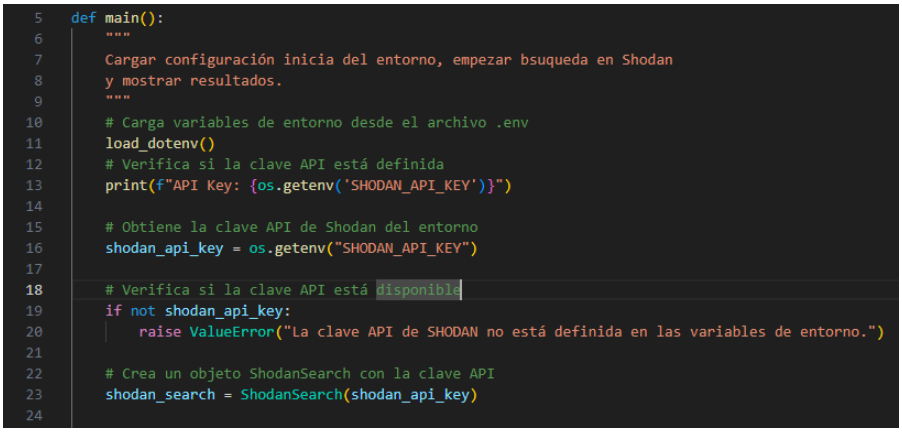
Parámetros del Script

Linea de código	Función
load_dotenv():	Carga las variables de entorno configuradas en el archivo .env El archivo .env, contiene configuración sensible, como la clave API, evitando que se ecriban directamente en el código.
{os.getenv('SHODAN_API_KEY')}")	Recibe el valor de la clave API de Shodan desde las variables del entorno. Es una practica recomendada para protección de credenciales.
Verificación de la clave.	Si la clave no esta determinada, se lanza un ValueError, deteniendo el script.
ShodanSearch(shodan_api_key)	Se invoca el objeto Shodan.search(), el cual devuelve un diccionario de información con los resultados de los activos expuestos en Internet.

Nota. La presenta tabla describe los parámetros de los módulos del script.

Figura 10

Script de Automatización para Búsqueda de Activos Expuesto a Internet.

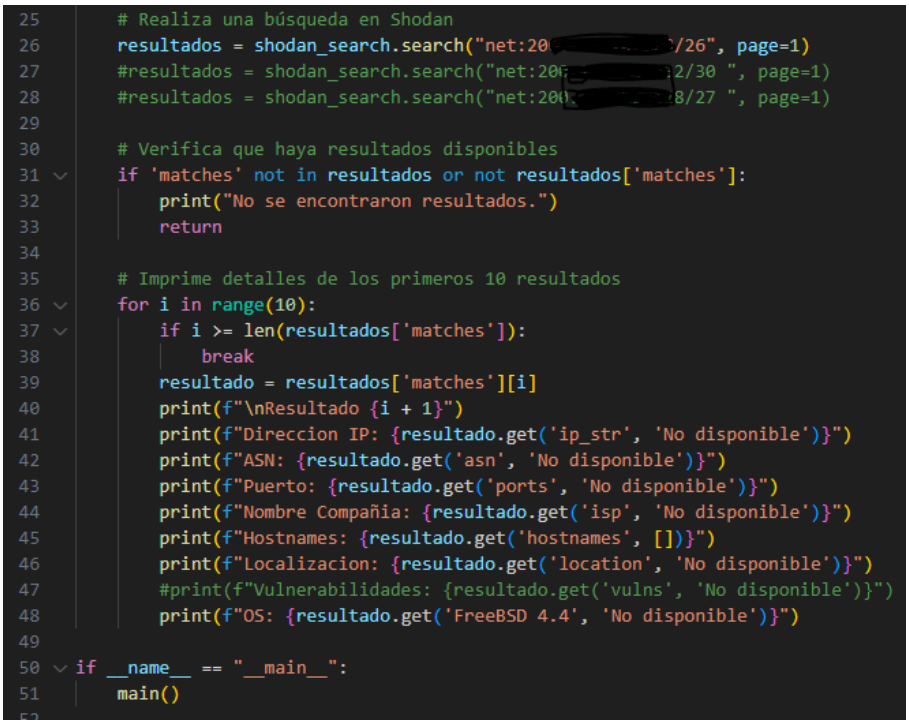


```python
def main():
    """
    Cargar configuración inicia del entorno, empezar bsuqueda en Shodan
    y mostrar resultados.
    """
    # Carga variables de entorno desde el archivo .env
    load_dotenv()
    # Verifica si la clave API está definida
    print(f"API Key: {os.getenv('SHODAN_API_KEY')}")

    # Obtiene la clave API de Shodan del entorno
    shodan_api_key = os.getenv("SHODAN_API_KEY")

    # Verifica si la clave API está disponible
    if not shodan_api_key:
        raise ValueError("La clave API de SHODAN no está definida en las variables de entorno.")

    # Crea un objeto ShodanSearch con la clave API
    shodan_search = ShodanSearch(shodan_api_key)

```

Nota. Configuración del script para descubrir activos en Internet.

Figura 11

Script de Automatización para Búsqueda de Activos Expuestos en Internet.



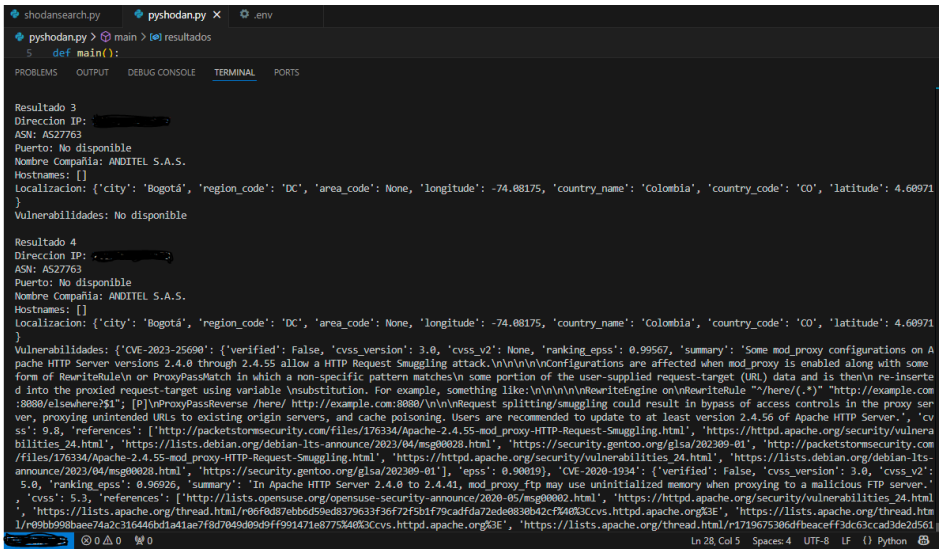
```python
    # Realiza una búsqueda en Shodan
    resultados = shodan_search.search("net:20[redacted]/26", page=1)
    #resultados = shodan_search.search("net:20[redacted]2/30 ", page=1)
    #resultados = shodan_search.search("net:20[redacted]8/27 ", page=1)

    # Verifica que haya resultados disponibles
    if 'matches' not in resultados or not resultados['matches']:
        print("No se encontraron resultados.")
        return

    # Imprime detalles de los primeros 10 resultados
    for i in range(10):
        if i >= len(resultados['matches']):
            break
        resultado = resultados['matches'][i]
        print(f"\nResultado {i + 1}")
        print(f"Direccion IP: {resultado.get('ip_str', 'No disponible')}")
        print(f"ASN: {resultado.get('asn', 'No disponible')}")
        print(f"Puerto: {resultado.get('ports', 'No disponible')}")
        print(f"Nombre Compañia: {resultado.get('isp', 'No disponible')}")
        print(f"Hostnames: {resultado.get('hostnames', [])}")
        print(f"Localizacion: {resultado.get('location', 'No disponible')}")
        #print(f"Vulnerabilidades: {resultado.get('vulns', 'No disponible')}")
        print(f"OS: {resultado.get('FreeBSD 4.4', 'No disponible')}")

if __name__ == "__main__":
    main()
```

Nota. Configuración del script para descubrir activos en Internet.

Ejecución del script.

Figura 12

Resultados de la Búsqueda en Shodan con el Script de Automatización.



Fuente. Resultados obtenidos del script de automatización.

El script recorre los rangos CIDR definidos previamente en la Fase 1, identificando un Total de:

- 9 direcciones IP expuestas en Internet.
- 3 direcciones IP con códigos CVE asociados.
- 1 direecion IPv4 carga directamente la interfaz web de un firewall interno.

Figura 13

Interfaz Web de Firewall Expuesto a Internet.

Nota. Activo descubierto con el script de automatización.

Al validar estas direcciones los respectivos Firewall, se determina que correponden las siguietes configuraciones:

- NAT 1:1
- VirtualIP segmento público.

Tabla 8

Vulnerabilidades Encontradas Recopilación Semi-Pasiva

CIDR	IP	CVE	DESCRIPCIÓN E IMPACTO DE VULNERABILIDAD
200.xx.xx.x28/26	200.xx.xxx.x57	CVE-2018-19052	Permiten divulgación de información del sistema. Su explotación puede facilitar la identificación de versiones vulnerables y apoyar fases posteriores de ataque contra la infraestructura de la organización.
	200.xx.xxx.x 38	CVE-2024-27316	Vulnerabilidad que puede permitir la ejecución de acciones no autorizadas o la divulgación de información dependiendo del servicio afectado. Representa un riesgo elevado debido a su actualidad y posible disponibilidad de exploits públicos.
	200.xx.xxx.x 57		Dispositivo expuesto sin CVE asociado indica la presencia de servicios accesibles públicamente que pueden ser utilizados para reconocimiento de red. Esto incrementa el vector de ataque del centro de datos.
	200.xx.xxx.x 44	CVE-2018-19052	Vulnerabilidad asociada con configuraciones inseguras o software desactualizado, que puede ser utilizada para recopilar información técnica del sistema y facilitar ataques dirigidos.
	200.xx.xxx.x 39		Dispositivo accesible durante la fase de recopilación semi-pasiva, lo que evidencia exposición de servicios sin controles restrictivos adecuados, aumentando el riesgo de reconocimiento por terceros no autorizados.

	200.xx.xxx.x 30	Activo detectado en red pública que puede ser objetivo de técnicas de fingerprinting. Su exposición puede ser objeto de ataques como, acceso no autorizado o ataques de denegación de servicio.
200.xx.xx.x52/30	200.xx.xxx.x 54	Activo detectado en red pública que puede ser objetivo de técnicas de fingerprinting. Su exposición puede ser objeto de ataques como, acceso no autorizado o ataques de denegación de servicio.
	200.xx.xxx.x 34	Activo con respuesta durante la recopilación semi-pasiva indica visibilidad externa de la infraestructura, lo cual incrementa el riesgo de ataques de reconocimiento.
	200.xx.xxx.x 32	Activo accesible públicamente facilita la enumeración de activos y posibles correlaciones con otras vulnerabilidades del entorno.
200.xx.xx.x28/26	200.xx.xxx.x 31	Activo accesible públicamente facilita la enumeración de activos y posibles correlaciones con otras vulnerabilidades del entorno.
	200.xx.xxx.x 34	Activo accesible públicamente facilita la enumeración de activos y posibles correlaciones con otras vulnerabilidades del entorno.
	200.xx.xxx.x 33	Activo accesible públicamente facilita la enumeración de activos y posibles

	correlaciones con otras vulnerabilidades del entorno.
200.xx.xxx.x 50	Activo accesible públicamente facilita la enumeración de activos y posibles correlaciones con otras vulnerabilidades del entorno.

Nota. La presente tabla describe las vulnerabilidades de los activos evaluados.

Recopilación Activa de la Información

Técnica de Escaneo

El presente proyecto se basa en la técnica de Pentesting Caja Blanca aplicada a la infraestructura en producción de seguridad de la empresa proveedora de servicios de Internet (ISP). Dada la sensibilidad de los activos evaluados (firewall, routers, y switchs en producción), es fundamental emplear técnicas de escaneo que cumplan con los siguientes requisitos:

- Minimice el impacto operativo**:** Garantizar que las pruebas no afecten la disponibilidad o rendimiento de los sistemas.
- Evite la generación de alertas: Prevenir la detección por parte de sistemas de firewall, IDS/IPS, durante el proceso de recopilación activa.
- Mantenga el sigilo: Asegurar que las actividades de escaneo pasen desapercibidas para no alertar a posibles atacantes o sistemas de monitoreo.

Fundamentos Del Protocolo TCP Para Escaneo De Red.

El protocolo TCP actúa en la capa de transporte y permite el envío de información de extremo a extremo detectando y corrigiendo errores garantizando la fiabilidad de los datos transmitidos. Antes de iniciar con el transporte de los datos, TCP realiza una negociación entre

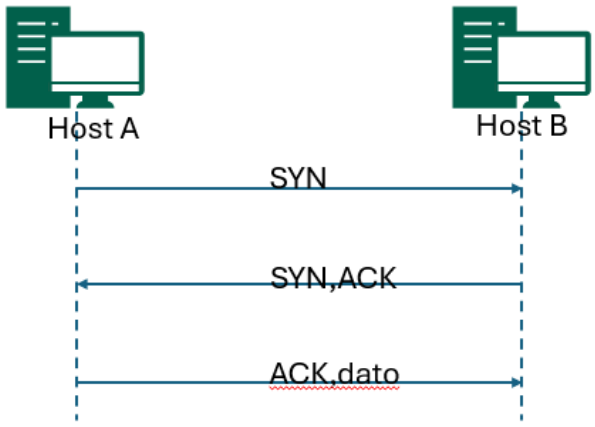
los equipos mediante el intercambio de 3 pasos "apretón de manos de tres vías" (3-way handshake).

Pasos de conexión TCP:

- Solicitud de conexión: el host que desea enviar datos (cliente), primero envía un segmento TCP con el bit SYN (Synchronize) al host receptor (servidor), indicando que sea establecer una conexión.
- El host receptor (servidor), si se encuentra disponible, responde con el segmento TCP que tiene le bit SYN activo y el bit ACK (Acknowledge) activado. También incluye el número de secuencia e indica que está listo para recibir los datos.
- Confirmación de conexión (ACK): el host cliente responde con un segmento TCP que tiene el bit ACK activado, confirmando que la conexión se ha establecido y se pueden enviar datos.

Figura 14

Proceso de Conexión Protocolo TCP



Nota. Fases de conexión del protocolo TCP

Técnica Half-Open (TCP-SYN Scan)

Esta técnica fue sido seleccionada debido a su capacidad para realizar escaneos masivos de puertos (miles por segundo) conservando un perfil discreto y sigiloso. Su eficacia radica en que no completa el proceso de establecimiento de conexiones TCP (three-way handshake), o que evita la generación de registros completos en los sistemas monitoreados y reduce significativamente la posibilidad de detección por mecanismos de seguridad

Principales características del método Half-open:

- Inicio de conexiones: Se envía exclusivamente un paquete con el flag SYN activado, correspondiente a la primera etapa del three-way handshake TCP
- Respuesta a puertos abiertos: Cuando un puerto está disponible, el host objetivo responde con un paquete SYN-ACK.
- Finalización controlada: El escáner no responde con ACK de confirmación, dejando la conexión a medio establecer (half-open).
- Ventaja de sigilo: Al no completarse el handshake TCP, no se generan entradas completas en los logs de conexión de los equipos objetivo.

Herramientas especializadas para escaneo de puertos TCP-SYN:

Nmap: (mapeador de redes) es un software open-source para a exploración de redes y auditoría de seguridad. Optimizado para el análisis rápido de infraestructuras a gran escala, también ofrece un rendimiento eficiente en equipos individuales (Guía de Referencia de Nmap, s.f.).

Scapy: Es una biblioteca de Python control granular sobre la manipulación de paquetes de red. Esta herramienta permite personalizar exhaustivamente cada parámetro del escaneo, lo que facilita la maximización del sigilo durante las pruebas (Scapy 2.6.1 Documentation, s.f.).

Estas herramientas representan soluciones estándar para implementar escaneos TCP-SYN: mientras Nmap es ampliamente adoptado en el ámbito de la ciberseguridad por su eficiencia, Scapy destaca por su flexibilidad para diseñar paquetes personalizados.

En la siguiente sección se presenta un análisis comparativo detallado de ambas herramientas, justificando la selección realizada para los requerimientos específicos de este proyecto de pentesting en entorno productivo.

Tabla 9

Comparación Herramientas de Escaneo de Red

Característica	Nmap	Scapy
Enfoque	Herramienta automatizada con técnicas de escaneo predefinidos (SYN, ACK, UDP, FIN).	Biblioteca Python para creación y manipulación personalizada de paquetes.
Detectabilidad	Mayor riesgo de detección (ej.: escaneos SYN rápidos generan logs en firewalls).	Bajo perfil: permite definir intervalos entre paquetes (*timing*) y modificar cabeceras para imitar tráfico legítimo.
Flexibilidad	Limitado a opciones preconfiguradas (--scan-delay, T1,--data-length) (-sS, -sT, -sU). No permite modificar la estructura de paquetes a bajo nivel.	Control total sobre capas 2-7 (Ethernet, IP, TCP/UDP).
Uso en infraestructuras monitorizadas	Alertas frecuentes en IDS/IPS por escaneos agresivos.	Escaneo más sigiloso al fragmentar paquetes o usar técnicas de low-and-slow.

Nota. La presente tabla describe las ventajas y desventajas de los programas de escaneo de red.

Dado que el objetivo de esta fase es identificar host y servicios expuestos sin generar tráfico agresivo ni actividades de escaneo convencionales, se ha seleccionado la herramienta Scapy por las siguientes ventajas estratégicas:

- Desarrollo de script personalizados: Permite implementar escaneos altamente sigilosos adaptados a los requerimientos específicos del proyecto.
- Control granular de paquetes: Ofrece gestión precisa sobre todos los parámetros de transmisión (timing, flags, TTL, fragmentación).
- Evasión de controles de seguridad: Reduce de forma significativa la probabilidad de detección por firewall/IDS configurados para bloquear patrones típicos de Nmap.
- Integración con el ecosistema Python: Facilita la conexión con otras fases del proyecto al utilizar el mismo lenguaje de programación.

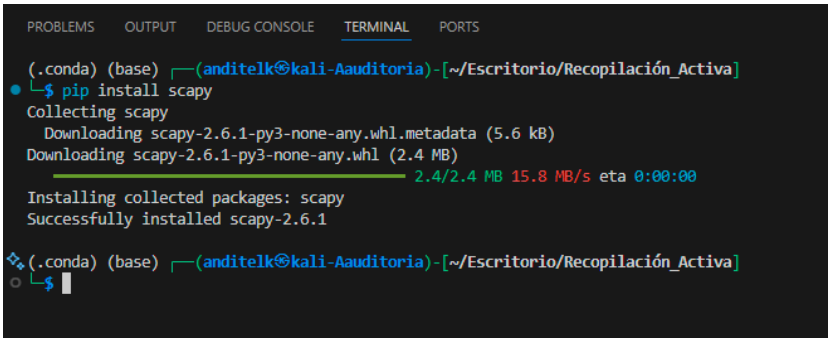
Esta selección garantiza el balance optimo entre eficacia en la recolección de información y minimización del impacto operativo en la infraestructura de seguridad evaluada.

Instalación de Scapy

Para la implementación de la técnica de escaneo, se procedió a instalar la biblioteca Scapy utilizando el gestor de paquetes PIP. La instalación se realiza dentro del entorno virtual Python previamente configurado, garantizando el aislamiento de dependencias y la reproducibilidad del ambiente en pruebas. Se ejecuto el siguiente comando:

Figura 15

Instalación de Librería Scapy

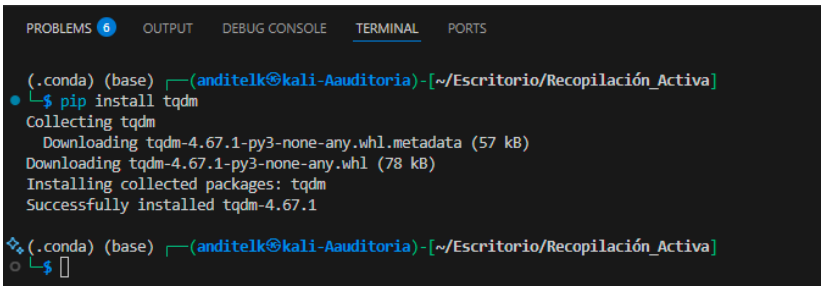


Nota. Proceso de instalación librería Scapy.

Se ejecuta la instalación de librerías adicionales que ayudan en la presentación de la información obtenida por Scapy.

Figura 16

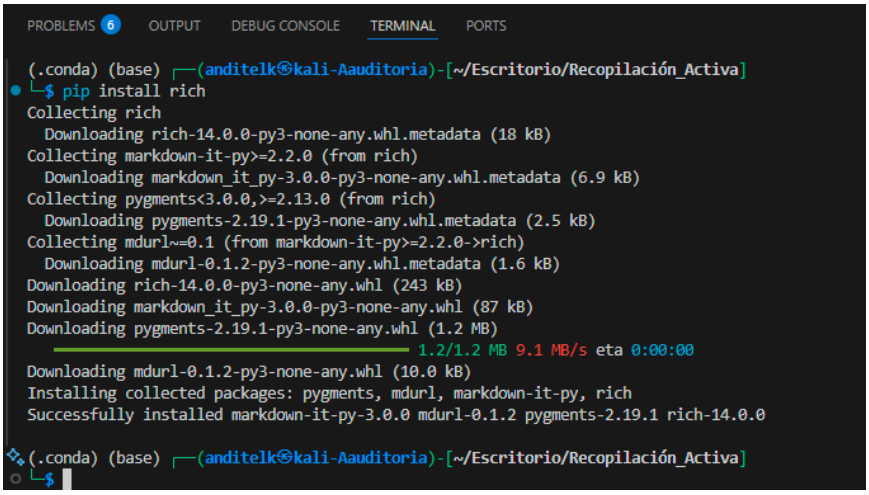
Instalación de Librería TDDM para Añadir Barra de Progreso Durante Ejecución del Script.



Nota. Proceso de instalación librería TDDM.

Figura 17

Instalación de Librería Rich, para Presentación de Datos en Tablas.



Nota. Proceso de instalación librería Rich.

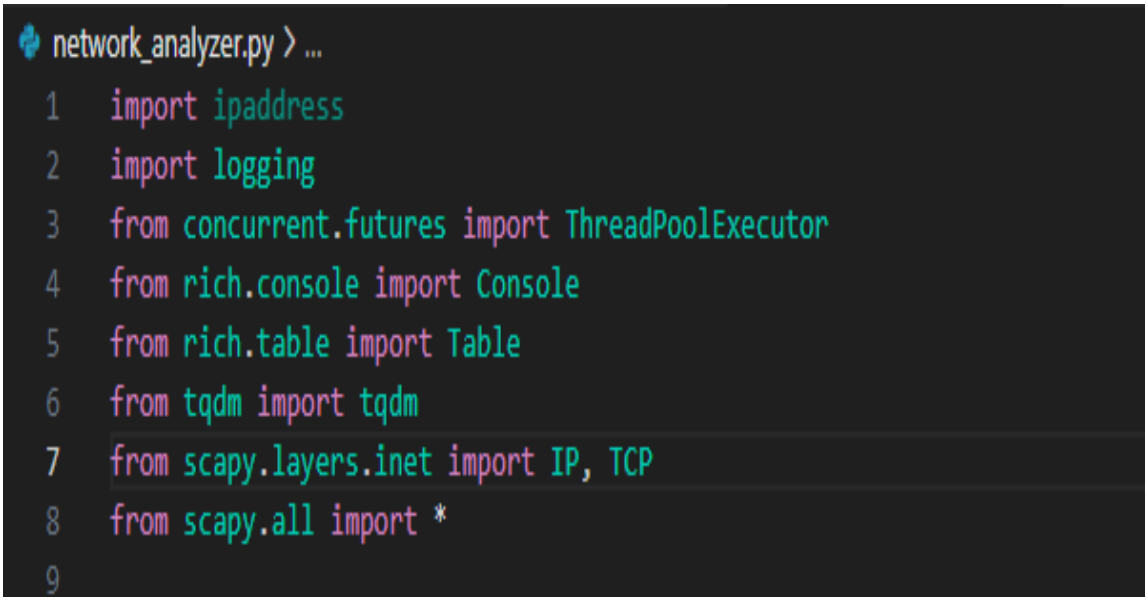
Escaneo de Host en Puertos Específicos

El script desarrollado contiene la clase Red_Datacenter_Seguridad, encargado de realizar el escaneo de puertos específicos en una lista de host por medio de paquetes TCP SYN, usando la librería Scapy.

1. Importación de Librerías.

Figura 18

Librerías Necesarias para Ejecución del Script Escaneo de Host.



Nota. Librerías necesarias para la ejecución del script de escaneo de activos.

Las primeras líneas de código corresponden a las librerías importadas.

Tabla 10

Librerías Implementadas en Script Recopilación Activa

Ipaddres	Esta librería permite trabajar con rangos y subredes en formato CIDR
Logging	Se usa para la gestión del mensaje del sistema o errores.
ThreadPoolExecutor	Permite la ejecución de tareas en forma paralela (multithreading).
rich.console y rich.table	Permite mostrar resultados en forma de tabla con formato amigable.
Tqdm	Añade barras de progreso durante la ejecución, de gran utilidad cuando se escanea CIDR.
Scapy	Principal herramienta para la creación y envío de paquetes TCP/IP personalizados.

Nota. La presente tabla describe la función de las librerías implementadas del script.

2. Configurar Scapy´s logger

Figura 19

Desactivación del Registrador Automático de la Librería Scapy.

```
# Desactivar warnings de Scapy en el log
logging.getLogger("scapy.runtime").setLevel(logging.ERROR)
```

Nota. Desactivación del registrador automático.

Scapy configura un registrador automático mediante el módulo de Python logging, por defecto está configurado en modo Warning. Para evitar interferencias con la salida del programa lo desactivamos cambiando el mensaje a ERROR.

3. Clase Principal.

Figura 20

Definición de la Clase Principal del Script.

```
class Red_Datacenter_seguridad:
    """Análisis de red para identificar hosts activos dentro de un rango de IP especificado.
       o Listado de host previamente definidos.
    Atributos:
        network_range (str): Rango de red en notación CIDR a analizar.
        timeout (int): Tiempo máximo en segundos para esperar respuesta de cada host.
    """
```

Nota. Descripción de la clase principal en el script.

La línea de código define una clase para el análisis de la red Datacenter, enfocado en el escaneo de los host y puertos previamente definidos en fases anteriores.

4. Constructor de Clase.

Figura 21

Definición del Constructor de Clase del Script.

```
def __init__(self, network_range, timeout=1):
    """Inicializa el analizador de red con el rango y el tiempo de espera especificados.

    Args:
        network_range (str): Rango de red en formato CIDR.
        timeout (int): Tiempo de espera para la respuesta de cada host, en segundos.
    """
    self.network_range = network_range
    self.timeout = timeout
```

Nota. Descripción del módulo constructor.

Ejecuta el analizador de red, con el rango de red y tiempo de espera por cada respuesta. El presente proyecto no se usa rango de red para no afectar la operación, se escanea una lista definida de dispositivos para escanear.

5. Escaneo TCP SYN con Scapy.

La siguiente línea de código, recibe la lista de direcciones IP y puertos del host objetivo.

Figura 22

Configuración de los Parámetros que Recibe la Librería Scapy para el Escaneo.

```
def _scan_host_scapy(self, ip, scan_ports=(80, 4443, 8443, 23)):
    """Se Implementa Scapy para escanear puertos específicos de un host utilizando paquetes TCP SYN.

    Args:
        ip (str): Dirección IP del host a escanear.
        scan_ports (tuple): Puertos a escanear en el host.

    Returns:
        tuple: Tupla conteniendo la IP del host y un booleano que indica si alguno de los puertos está abierto.
    """
```

Nota. Configuración de parámetros de red.

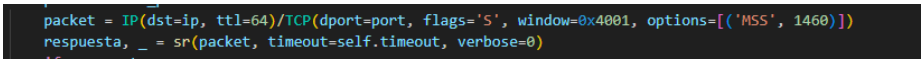
Se construye el paquete TCP/IP modificando los siguientes parámetros:

TTL=64: (TTL estándar) Evita detección como tráfico externo.

Windows=0x4001: Parámetro común en sistemas Windows/Linux para parecer legítimo.

Figura 23

Personalización del Paquete TCP.



Nota. Configuración de parámetros del paquete TCP.

La opción “sr” envía el paquete y espera respuesta.

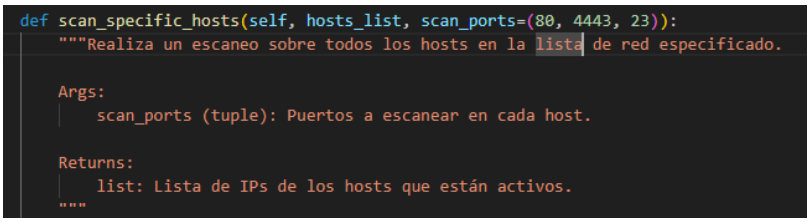
Si se recibe una respuesta, se considera que el host esta activo en ese puerto.

6. Escaneo de lista especifica de IPs.

La siguiente línea de código, se escanea la lista especifica de host, previamente establecidos en la fase de recopilación pasiva de la información.

Figura 24

Código para Leer Lista Especifica de los Activos a Escanear.



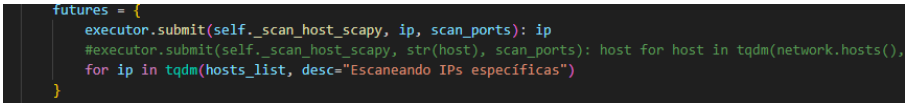
Nota. Descripción de tarea del código.

Las librerías ThreadPoolExecutor, es implementada para paralelizar las tareas, esta es una buena opción cuando se escanea rangos de red completos. La librería tqdm, se usa para mostrar el progreso.

La función de las siguientes líneas de código es envía tareas al método _scan_host_scapy, para cada una de las IP en la Lista. Si el resultado indica un puerto abierto, se agrega la IP a la lista de host_up

Figura 25

Configuración de la Librería Tqdm en el Script.



Fuente. Parámetros de configuración de Librería Tqdm.

7. Presentación de resultados en formato Tabla.

La siguiente línea de código presenta los resultados utilizando una tabla de la librería rich. Se define la columna de salida con estilo y cada ip detectada se añade en la fila.

Figura 26

Configuración de la Librería Pretty para la Presentación de Resultados.

```
def pretty_print(self, data, data_type="hosts"):
    """Imprime los datos de manera elegante en una tabla.

    Args:
        data (list): Datos a imprimir.
        data_type (str): Tipo de datos para adecuar la impresión.
    """
    console = Console()
    table = Table(show_header=True, header_style="bold magenta")
    if data_type == "hosts":
        table.add_column("Hosts Activos Datacenter", style="bold green")
        for host in data:
            table.add_row(host, end_section=True)
    console.print(table)
```

Nota. Parámetros de configuración Librería Pretty.

8. Script Main.

El script Main, se encarga de leer las direcciones IPs desde un archivo .txt, instanciar la clase de escaneo, ejecutar el análisis y presentar los resultados.

Figura 27

Configuración de Parámetros del Script Main.

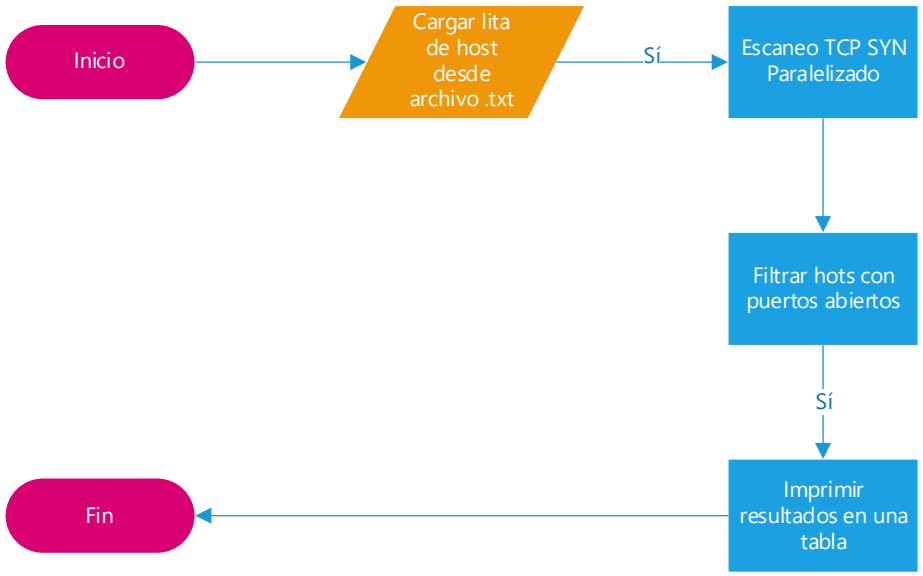
```python
from network_analyzer import Red_Datacenter_seguridad

with open("host_objetivo.txt", "r") as file:
    scan_specific_hosts = [line.strip() for line in file if line.strip()]


# Crear una instancia de la clase (se requiere pasar algún rango, aunque no lo uses)
analyzer = Red_Datacenter_seguridad('192.168.x.0/24')



# Ejecutar escaneo sobre IPs específicas
hosts_up = analyzer.scan_specific_hosts(scan_specific_hosts)

# Mostrar resultados
analyzer.pretty_print(hosts_up)
```

Nota. Definición de parámetros script Main.

9. Flujo de trabajo.

Figura 28

Flujo de Trabajo Recopilación Activa.



Nota. Flujo de trabajo del script.

10. Ejecución de script main.py

Figura 29

Resultados del Script Escaneo de Host en Puertos Específicos.

Nota. Dispositivos descubiertos.

Listado de dispositivos escaneados en los puertos (80, 4443, 8443, 23)

Tabla 11

Listado de Activos Encontrados en Puertos (80, 4443, 8443, 23)

Hosts Activos Datacenter
192.168.1.xx6
200.25.2xx.xx0
200.25.2xx.xx0
200.25.2xx.xx4
192.168.xx.xx6
192.168.xx.xx4
192.168.xx.xx0
192.168.xx.xx1
192.168.xx.xx2
192.168.xx.xx4
192.168.xx.xx3
10.100.xx.xx0
192.168.xx.xx
200.25.2xx.xx
200.25.2xx.xxx

Nota. La presenta tabla presenta listado de activos descubiertos.

Escaneo de Puertos

Para optimizar el análisis de puertos en la infraestructura de red del centro de datos evaluado, se integró al script del proyecto el módulo socket de Python, junto con un nuevo método llamado ports_scan, el cual realiza escaneo eficiente y preciso de puertos abiertos en los activos previamente definidos.

Módulo Soket. El módulo socket, parte de la librería estándar de Python, permite la comunicación directa mediante el protocolo TCP/IP. En el contexto de este proyecto, se implementó para validar si un puerto especifico está en estado de escucha (listening) enuna dirección IP determinada, mediante la función connect_ex, la cual opera de la siguiente manera:

- Establecimiento de la conexión: Intenta establecer la conexión a un puerto especifico en la IP objetivo.

- Retorno de valores:

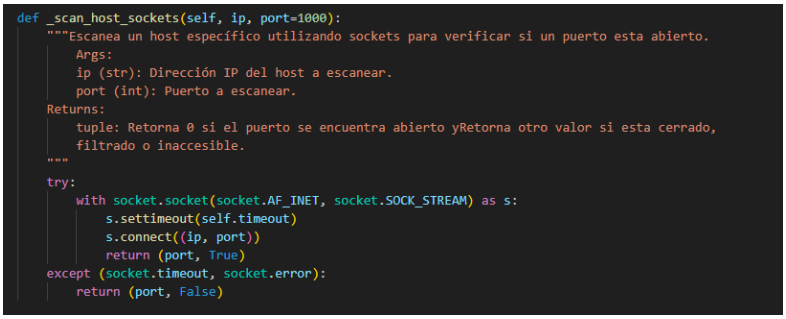
0: Indica que el puerto está abierto y accesible.

Otros valores: Señalan que el puerto está cerrado, filtrado por un Firewall o inaccesible.

Este tipo de escaneo es considerado no intrusivo al no enviar cargas útiles complejas ni ejecuta negociaciones de servicios, reduciendo el riesgo de interrupciones en ambientes de red en producción y siguiendo la técnica de pentesting Caja Blanca.

Figura 30

Integración del Módulo Socket al Script.



```python
def _scan_host_sockets(self, ip, port=1000):
    """Escanea un host específico utilizando sockets para verificar si un puerto esta abierto.
        Args:
        ip (str): Dirección IP del host a escanear.
        port (int): Puerto a escanear.
    Returns:
        tuple: Retorna 0 si el puerto se encuentra abierto yRetorna otro valor si esta cerrado,
        filtrado o inaccesible.
    """
    try:
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.settimeout(self.timeout)
            s.connect((ip, port))
            return (port, True)
    except (socket.timeout, socket.error):
        return (port, False)
```

Nota. Configuración Modulo Socket

Método Ports_Scan. La implementación del método ports_scan al script permite:

- Escaneo dirigido: Analizar una lista especifica de direcciones IP, predefinidas en la fase de interacciones previas, garantizando que el alcance del proyecto se mantenga alineado con los activos críticos identificados.

- Evaluación paralela: Realizar comprobaciones simultaneas de los puertos definidos en la Fase 1 junto con un rango extendidos (puertos 1-9000.), optimizando el tiempo de ejecución mediante técnicas de multihilo.

- Gestión estructurada de resultados: Almacenar los hallazgos en un diccionario (dict), donde cada IP se asocia a los puertos que se encuentran abiertos.

Figura 31

Integración del Método Port Scan al Script.

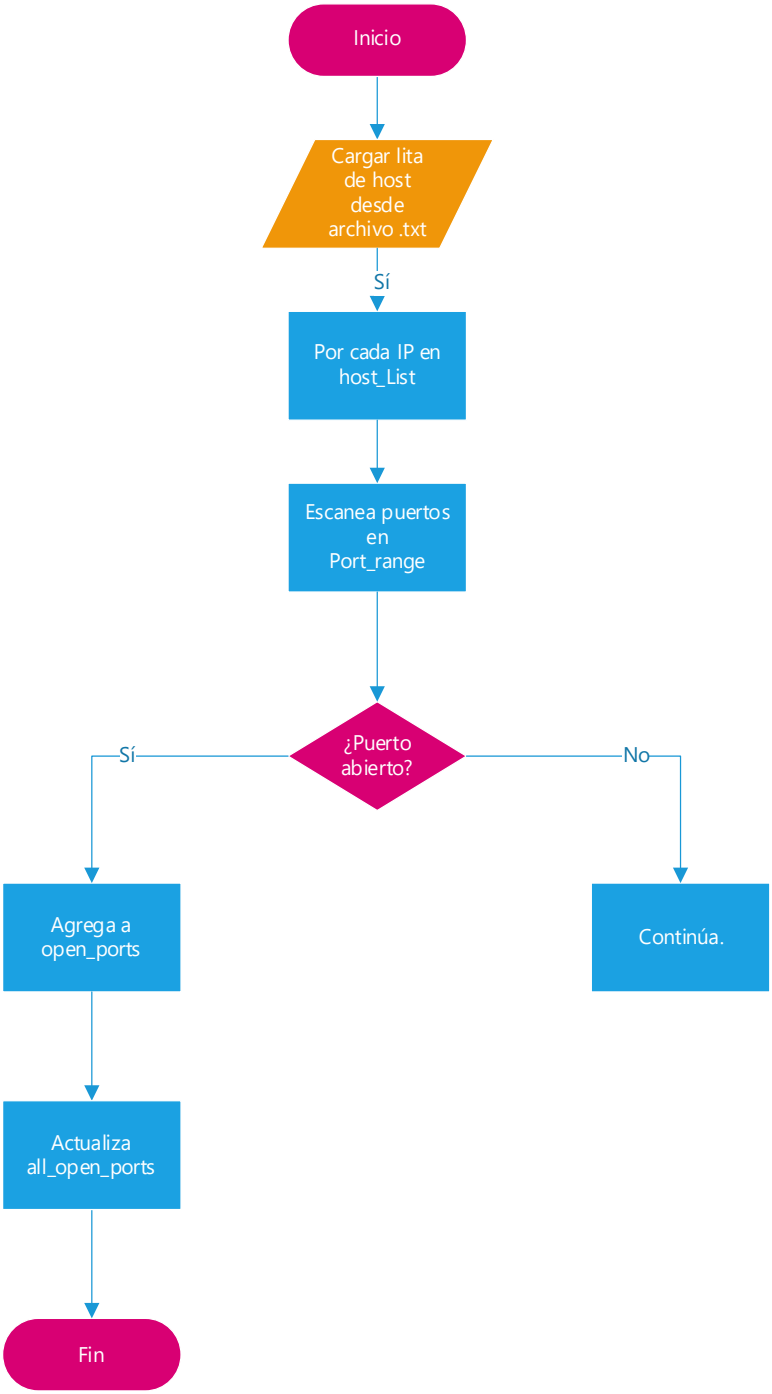
```
def ports_scan(self, hosts_list, port_range=(1, 1025)):
    #active_host = self.scan_specific_hosts()
    all_open_ports = {}
    for ip in hosts_list:
        open_ports = []
        for port in tqdm(range(*port_range), desc=f"Escaneando puertos en {ip}"):
            _, is_open = self._scan_host_scapy(ip, [port])
            if is_open:
                open_ports.append(port)
        if open_ports:
            all_open_ports[ip] = open_ports
    return all_open_ports
```

Nota. Configuración modulo port scan.

Flujo de trabajo.

Figura 32

Flujo de Trabajo Modulo Escaneo de Puertos.



Nota. Descripción del flujo de trabajo del script.

Ejecución del script para escaneo de puertos.

Figura 33

Resultados de Escaneo de Puertos.

Nota. Activos encontrados en puertos 21,23,53 y 80.

Listado de host y puertos abiertos.

Tabla 12

Resultado de Puertos Descubiertos en Activos de Red

Dirección IP	Puertos Abiertos
192.168.xx.xx6	23, 80
200.25.2xx.xx0	53, 80
200.25.2xx.xx0	53, 80
192.168.xx.xx6	21, 23, 80
192.168.xx.xx4	21, 23, 80
192.168.xx.xx0	21, 23, 80
192.168.xx.xx1	21, 23
192.168.xx.xx2	21, 23
192.168.xx.xx3	23
10.100.xx.xx0	21, 23, 80

Nota. La presente tabla se relaciona las IP con los puertos abiertos.

Escaneo de Servicios

El escaneo de servicios se fundamenta en la captura de los banners, los cuales brindan metada critica para la identificación de versiones de software. Esta técnica se implementa mediante conexiones TCP con socket, permitiendo detectar servicios vulnerables mediante el análisis de respuesta a estímulos controlados (mensajes genéricos).

Método get_banner(self, ip, port)

El propósito de este método es establecer una conexión TCP directa a un puerto abierto y capturar el banner de servicio, el cual contiene metadatos con versiones de software o mensajes de identificación.

Implementación:

Tabla 13

Características Del Método Get_Banner

Conexión TCP	Implementa la librería Socket para conectarse al puerto objetivo.
Solicitud Genérica.	Envío de mensaje Hello\r\n para solicitar una respuesta del servicio.
	Búfer ampliado: 6144 bytes para captura de respuesta extensas.
Configuraciones robustas.	Manejo de errors: error=" ignore" evita fallos en la codificación de caracteres no estándar.

Figura 34

Integración del Módulo Get Banner al Script.

```
def get_banner(self, ip, port):
    try:
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.settimeout(self.timeout + 20)
            s.connect((ip, port))
            s.send(b'Hello\r\n')
            #return s.recv(1024).decode().strip()
            return s.recv(6144).decode(errors="ignore").strip()
    except Exception as e:
        return f"Error: {e}"
        #return str(e)
```

Nota. Configuración del método get.

Método services_scan (self, hosts_ports_dict)

Este método realiza la orquestación del escaneo de servicios y utiliza la función get_banner para todos los puertos abiertos previamente detectados en un diccionario del tipo {IP: [puertos]}.

Para optimizar el rendimiento, el escaneo se realiza en paralelo implementando ThreadPoolExcutor con hasta 100 hilos simultáneos

Los resultados se almacenan en una estructura tipo diccionario {IP: {puerto: banner}}, que posteriormente puede ser usada para visualizar.

Finalmente se filtran las respuestas vacías o errores típicos como timed out, refuse o No rute Host, conservando únicamente los banners útiles.

Figura 35

Integración del Módulo Service Scan al Script.

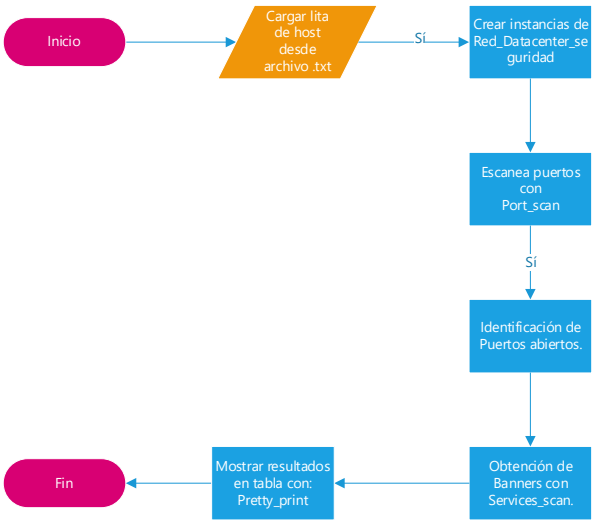
```
def services_scan(self, hosts_ports_dict):

    #active_hosts = self.scan_specific_hosts()
    services_info = {}
    with ThreadPoolExecutor(max_workers=100) as executor:
        for ip, ports in hosts_ports_dict.items():
        #for ip in active_hosts:
            futures = []
            services_info[ip] = {}
            for port in tqdm(ports, desc=f"Obteniendo banners en {ip}"):
            #for port in tqdm(range(*port_range), desc=f"Obteniendo banners en {ip}"):
                future = executor.submit(self.get_banner, ip, port)
                futures.append((future, port))
            for future, port in futures:
                result = future.result()
                if result and 'timed out' not in result and 'refused' not in result and 'No route to host' not in result:
                    services_info[ip][port] = result
    return services_info
```

Nota. Configuración del módulo service scan.

Flujo de trabajo.

Figura 36

Flujo de Trabajo Módulo Escaneo de Servicios.



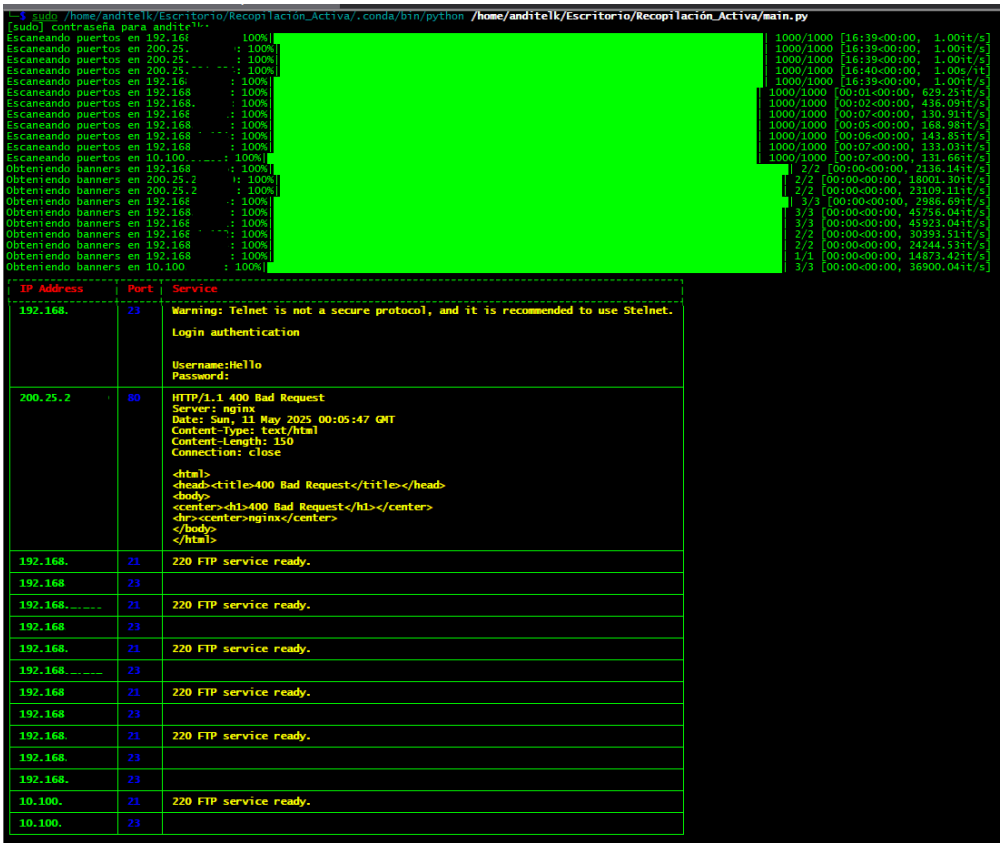
Nota. Flujo de trabajo del script.

Ejecución del script y resultados.

Se ejecuta el archivo main.py, obteniendo los siguientes resultados:

Figura 37

Resultado de Ejecución de Script para Obtener Banners.



Nota. Banners descubiertos.

Tabla 14

Resultados del Script Recopilación Activa

IP Address	Port	Service
192.168.1.xx6	23	Warning: Telnet is not a secure protocol, and it is recommended to use Stelnet. Login authentication
200.25.2xx.xx0	80	HTTP/1.1 400 Bad Request Server: nginx Date: Mon, 12 May 2025 20:56:03 GMT Content-Type: text/html Content-Length: 150 Connection: close <html> <head><title>400 Bad Request</title></head> <body> 400 Bad Request <hr> nginx </body> </html>
192.168.x.xx4	21	220 FTP service ready.
192.168.x.xx4	23	
192.168.x.xx0	21	220 FTP service ready.
192.168.x.xx0	23	
192.168.x.xx1	21	220 FTP service ready.
192.168.x.xx1	23	
192.168.x.xx2	21	220 FTP service ready.
192.168.x.xx2	23	
192.168.x.xx4	21	220 FTP service ready.
192.168.x.xx4	23	
192.168.x.xx3	23	
10.100.x.xx0	21	220 FTP service ready.
10.100.x.xx0	23	

Nota. Banners descubiertos en los activos de la infraestructura.

Modelado de Amenazas

La fase de modelado de amenazas es fundamental en la prueba de pent-testing, tanto para el probador, como para la organización, ya que permite identificar, clasificar y priorizar las potenciales amenazas a las que se enfrenta la infraestructura analizada.

En esta etapa, se adopta la metología PTES (Penetration Testing Execution Standard), estructurando el análisis de acuerdo con las siguientes fases:

Metodología

Se adopta un enfoque basado en los pasos propuestos por la metodología PTES:

Tabla 15

Metodología para Modelado de Amenazas

Elemento	Descripción
Recolección de la Información.	Desarrollada en las fases de recopilación Pasiva, semi-pasiva y activa.
Identificación y clasificación de los activos.	Se realiza clasificación de los activos primarios y secundarios de acuerdo con su criticidad en la infraestructura.
Identificación comunidad de amenazas.	Se analiza las amenazas de actores externos (hackers, malware, botnets) y actores internos (empleados, errores de configuración)
Correlación de amenazas.	Implementación de la herramienta Microsoft Threat Modeling Tool, para la visualización del flujo de datos entre componentes y servicios.

Nota. La presente tabla describe la metodología para el modelo de amenazas.

Recolección de la Información

Los activos fueron definidos previamente en el inventario de la fase pasiva y verificados a través de las fases semi-pasiva y activa mediante técnicas como escaneo TCP-SYN y captura de banners.

Identificación y Clasificación de los Activos

Tabla 16

Clasificación de los Activos

ID	Activo	Tipo	IP Pública/Privada	Prioridad
1	Firewall Core	Hardware	Pública	Alta
2	Servidor VPN	Hardware	Pública	Alta
3	Switch Distribución	Hardware	Privada	Media
4	Router Core	Hardware	Pública	Alta
5	Switch Acceso	Hardware	Privada	Baja

Nota. La presente tabla se relaciona los activos clasificados.

Identificación de la Comunidad de Amenazas

Tabla 17

Clasificación de Amenazas

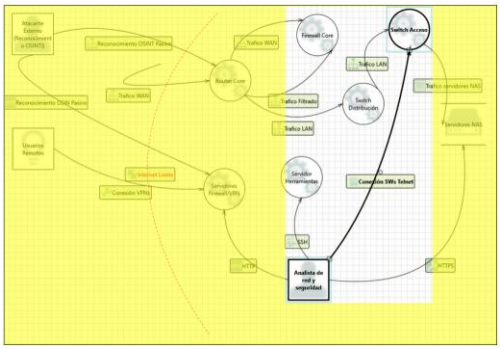
Externas	Ciberdelincuentes buscando servicios expuestos en Internet mediante técnicas OSINT.
	Botnets automatizados escaneando puertos comunes (21,22,23,80,443)
	Ataques dirigidos a vulnerabilidades conocidas (CVE detectados.)
Internas	Empleados desinformados o malintencionados.
	Errores de configuración en Firewall, switch, router o VPN.
Ambientales	Fallas en suministro eléctrico sin respaldo.
	Ausencia de políticas de respaldo y redundancia.

Nota. La presente tabla se clasifica las amenazas.

Correlación de Amenazas

Figura 38

Diagrama Modelado de Amenazas.



Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

Se implementa la herramienta Microsoft Thread Modeling Tool para visualizar el flujo de datos entre los componentes y servicios. El reporte arroja 44 amenazas en total, categorizado de la siguiente manera:

Tabla 18

Resultados Correlación de Amenazas

Etiquetas de fila	Cuenta de Category
Denial Of Service	8
Elevation Of Privilege	17
Information Disclosure	4
Repudiation	3
Spoofing	8
Tampering	3
User-defined	1
Total, general	**44**

Nota. *Resultados obtenidos por Microsoft Tharead Modeling Tool.*

Amenazas asociadas en cada interacción:

Interacción conexión SWs Telnet

Tabla 19

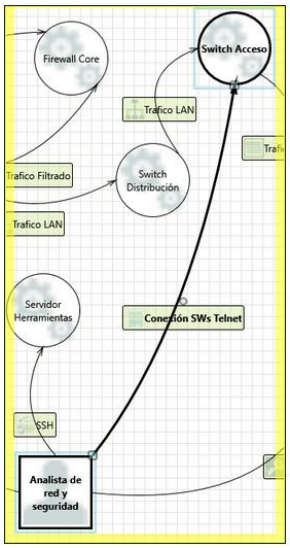
Resultado Interacción SWs y Telnet

Conexión SWs Telnet	5
Denial Of Service	1
Elevation Of Privilege	1
Information Disclosure	1
Spoofing	1
Tampering	1

Nota. Resultado Interacción SWs y Telnet

Figura 39

Resultado Interacción SWs, Telnet.

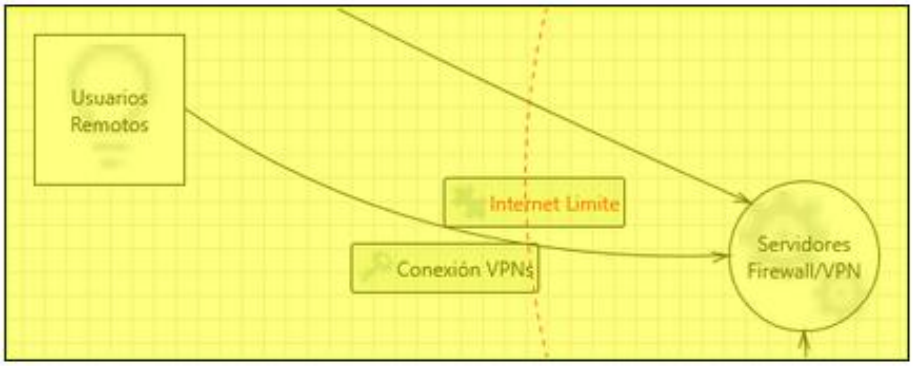


Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

1. Conexión VPNs

Figura 40

Resultado Interacción VPNs



Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

Tabla 20

Resultado de Interacción VPN

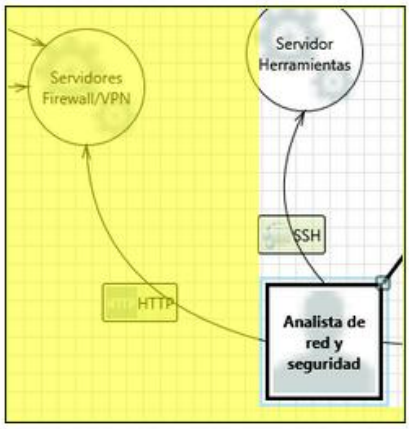
Conexión VPNs	6
Denial Of Service	2
Elevation Of Privilege	3
Repudiation	1

Nota. Resultado de Interacción VPN

2. Interacción HTTP.

Figura 41

Resultado Interacción Protocolo HTTP.



Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

Tabla 21

Resultado Interacción Protocolo HTTP

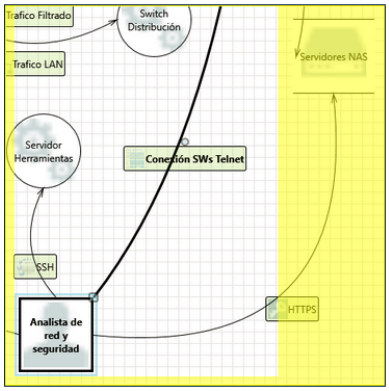
HTTP	2
Elevation Of Privilege	1
Spoofing	1

Nota. Interacción Protocolo HTTP

3. Interacción HTTPS.

Figura 42

Resultado Interacción Protocolo HTTPS



Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

Tabla 22

Resultado Interacción Protocolo HTTPS

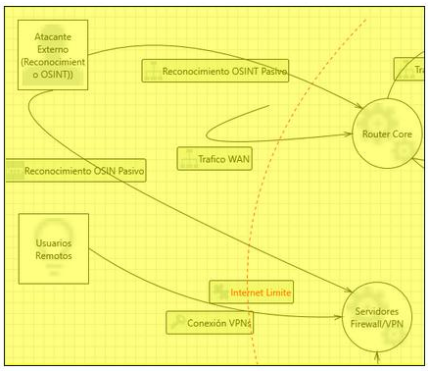
HTTPS	1
Spoofing	1

Nota. Interacción Protocolo HTTPS

4. Reconocimiento OSINT Pasivo.

Figura 43

Resultado Interacción OSINT Pasivo.



Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

Tabla 23

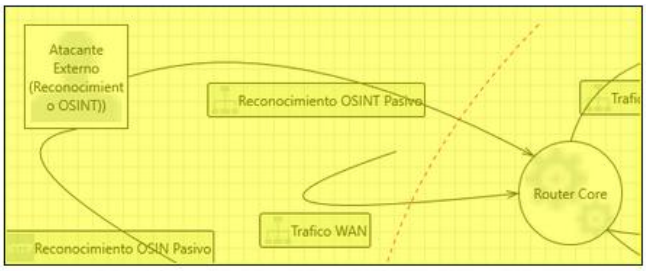
Resultado Interacción OSINT Pasivo

Reconocimiento OSIN Pasivo	10
Denial Of Service	2
Elevation Of Privilege	3
Information Disclosure	1
Repudiation	1
Spoofing	2
Tampering	1

Nota. Interacción OSINT Pasivo

Figura 44

Resultado Interacción OSINT Pasivo



Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

Tabla 24

Resultado Interacción OSINT Pasivo

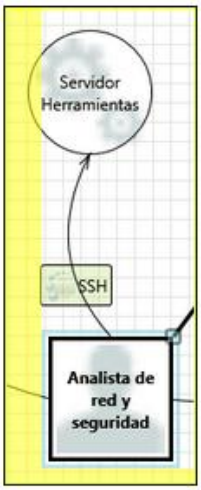
Reconocimiento OSINT Pasivo	
Denial Of Service	2
Elevation Of Privilege	4
Information Disclosure	1
Repudiation	1
Spoofing	2
Tampering	1

Nota. Resultados de Interacción

5. SSH.

Figura 45

Resultado Interacción Protocolo SSH.



Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

Tabla 25

Resultado Interacción Protocolo SSH

SSH	1
Elevation Of Privilege	1

Nota. Resultados de Interacción Protocolo SSH

6. Trafico filtrado.

Tabla 26

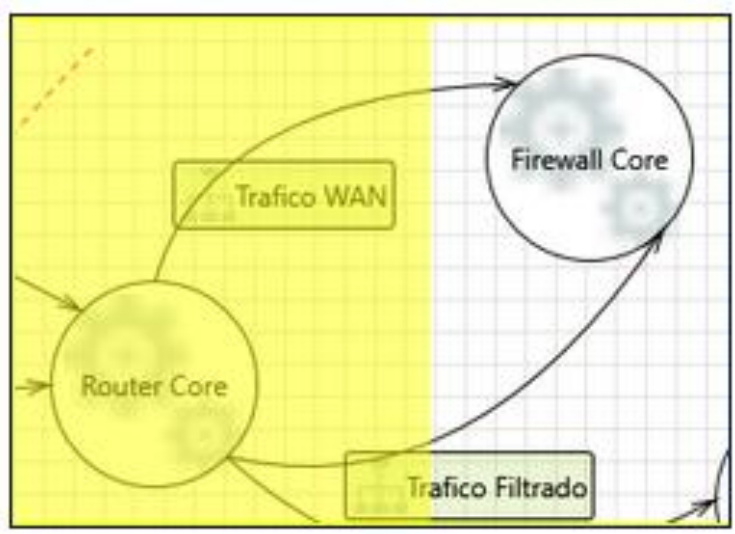
Resultado Interacción Filtrado Tráfico

Trafico Filtrado	1
Elevation Of Privilege	1

Nota. Resultados de Interacción Filtrado de Tráfico

Figura 46

Resultado Interacción Tráfico Filtrado.

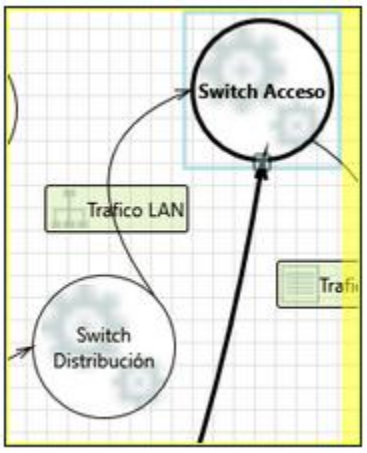


Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

7. LAN tráfico.

Figura 47

Resultado Interacción Tráfico LAN.



Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

Tabla 27

Resultado Interacción LAN

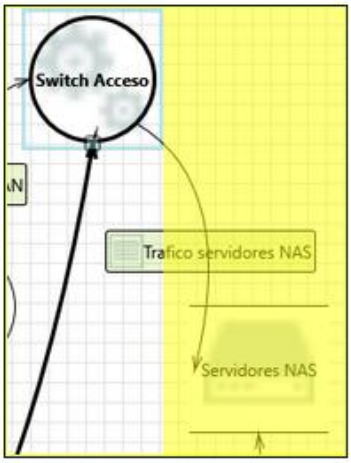
Trafico LAN	2
Elevation Of Privilege	2

Nota. Resultados de Interacción LAN

8. Trafico Servidores NAS.

Figura 48

Resultado Interacción Servidores.



Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

Tabla 28

Resultado Interacción NAS

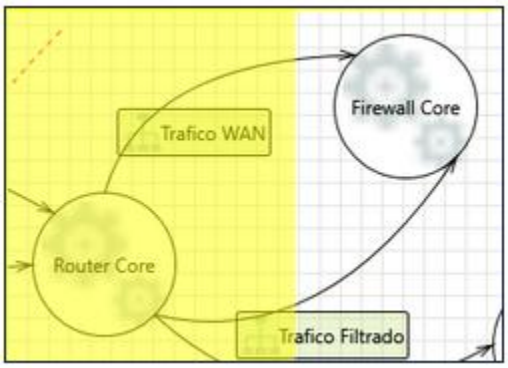
Trafico servidores NAS	2
Denial Of Service	1
Spoofing	1

Nota. Resultados de Interacción NAS

9. Trafico WAN.

Figura 49

Resultado Interacción WAN.



Nota. Captura de pantalla Modelado de Amenazas por Microsoft Tharead Modeling Tool.

Tabla 29

Resultado Interacción WAN

Trafico WAN	1
Elevation Of Privilege	1

Nota. Resultados de Interacción WAN

Mitigaciones a Amenazas

Denial of service (DoS).

Interrupción de la disponibilidad de un sistema o servicio por medio de ataques que saturan recursos (ancho de banda, CPU, memoria.)

Mitigaciones:

- Firewall y sistemas IDS/IPS:
  - Configurar reglas para bloquear tráfico malicioso (SYN floods, ICMP)

- Rate limiting y control de recursos:
  - Limitar el número de solicitudes por segundo y protege contra abusos mediante timeout, cuotas y balanceadores de carga.

Elevation of Privilege (EoP).

Un atacante explota vulnerabilidades para escalar privilegios.

Mitigaciones:

- Principio de menor privilegio:
  - Segmentar cuentas: Crear cuentas de usuario estándar (sin acceso root) y cuenta de administradores (Acceso controlado por MFA)
  - Validación estricta de roles:
  - Autenticación multifactorial (MFA): implementar herramientas como Google Authenticator, YubiKey o certificados digitales.

Information Disclosure.

Exposición no autorizada de datos sensibles.

Mitigaciones:

- Cifrado en tránsito y en reposo:
  - Implementar TLS/SSL para comunicaciones, cifrado de discos y bases de datos.
- Control de acceso basado en roles (RBAC):
  - Crear políticas de usuarios autorizados para el acceso a información confidencial.

Repudiation.

Incapacidad de rastreo de accesiones a un usuario especifico, facilitando negación y actividades maliciosas.

Mitigaciones:

- Registros auditables y firmados digitalmente:
  - Centralizar logs con herramientas SIEM (Splunk, Elastic Stack)
- Autenticación y autorización fuerte:
  - Asegura que todas las operaciones puedan rastrearse hacia una identidad única.

Spoofing.

Falsificación de identidad como direcciones (IP, MAC, usuarios), para acceder a sistemas.

Mitigaciones:

- Autenticación Multifactor (MFA):
  - Implementar FIDO2 (llaves físicas) o biometría (huella digital)
- Validación criptográfica de identidad:
  - Implementación de certificados digitales o tokens firmados.

Tampering.

Manipulación no autorizada de datos (Alteración de configuraciones, inyección de código)

Mitigaciones:

- Integridad de datos mediante hashing:
  - Implementar SHA-256, HMAC o similares.

Modelo STRIDE

Para fortalecer el análisis de amenazas, se implementa el modelo STRIDE, desarrollado por Microsoft. Este marco metodológico clasifica las amenazas en seis categorías principales:

Tabla 30

Modelo STRIDE

Categoría	Descripción	Ejemplo
Spoofing	Suplantación de identidad	Un actor malicioso Externo/Interno se hace pasar por un router o servidor legitimo.
Tampering	Manipulación de datos.	Modificación no autorizada de configuraciones del Firewall.
Repudiaton	Repudio a acciones.	Un usuario niega haber realizado una acción maliciosa.
Information Disclosure	Divulgación de Información.	Filtración de información sensible por interfaces públicas.
Denial of Service	Denegación de servicio.	Saturación de routers o firewall que cause caída en los servicios.
Elevation of Privilege	Elevación de Privilegio.	Un atacante obtiene permisos administrativos no autorizados.

Nota. La presente tabla muestra la descripción del modelo STRIDE.

Análisis de STRIDE sobre Activos Críticos

Tabla 31

Análisis STRIDE sobre Activos

Activo	**S**	**T**	**R**	**I**	**D**	**E**
Firewall Core	✓	✓	X	✓	✓	✓
Router Core	✓	✓	X	✓	✓	✓
Servidor VPN Mikrotik	✓	X	X	✓	✓	✓
Switch Distribución	X	✓	X	X	✓	X

Nota. Amenaza potencial identificada, X amenaza con bajo riesgo.

Identificar las Vulnerabilidades en la Infraestructura

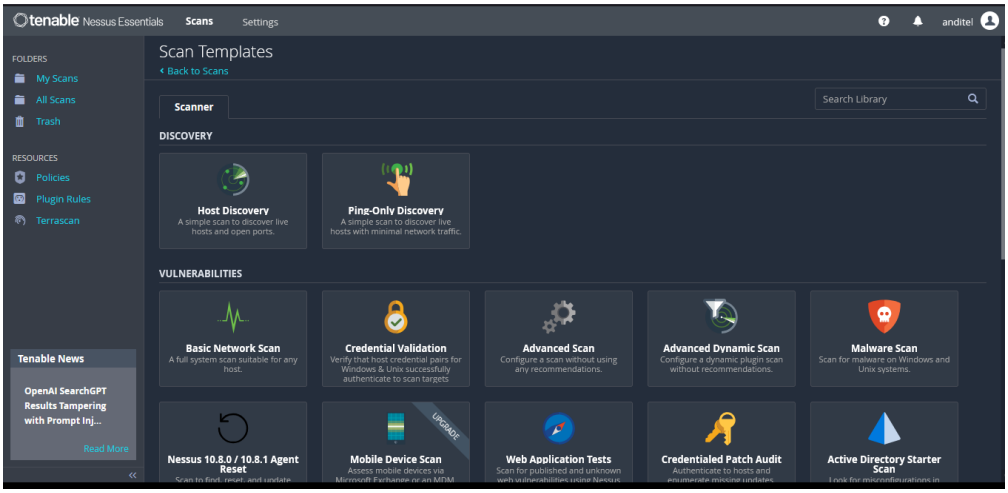
Esta fase ejecuta el descubrimiento de fallas en sistemas, configuraciones o aplicaciones que pueden ser aprovechadas por actores maliciosos (Internos y Externos) para comprometer la infraestructura. Estas fallas provienen desde configuraciones erróneas en los dispositivos y servicios, diseño de aplicaciones inseguras, software sin soporte oficial o falta de actualizaciones.

Herramienta de Escaneo Nessus.

Para el desarrollo de esta fase se opta por la implementación del software Nessus versión Essentials, debido a sus capacidades avanzadas de escaneo de vulnerabilidades y su amplia aceptación en la industria de ciberseguridad que de acuerdo con CRN, posiciona a Tenable fabricante de Nessus, como una de las 20 principales empresas de seguridad.

Figura 50

Interfaz Web Software Nessus



Fuente. Interfaz del software Nessus.

Principales Características de Nessus

- Amplia base de datos de vulnerabilidades: Nessus, es actualizado de forma constante con nuevas firmar y definiciones de vulnerabilidades basadas en CVE (Common Vulnerability Scoring System), facilitando la priorización del tratamiento de riesgos.
- Clasificación basada en CVSS: permite organizar las vulnerabilidades encontradas de acuerdo con su severidad (baja, media, alta o critica), utilizando el estándar CVSS (Common Vulnerability Scoring System), facilitando la priorización del tratamiento de riesgos.
- Interfaz amigable y generación de reportes detallados: Nessus, brinda informes técnicos que documentan claramente los hallazgos, incluyendo la descripción de la vulnerabilidad, ruta de explotación, sistemas afectados y recomendaciones de mitigación.
- Compatibilidad y flexibilidad: La herramienta permite el escaneo de múltiples tipos de dispositivos y sistemas operativos, adaptándose a la variedad de los activos del centro de datos, entre los cuales se encuentran firewalls, routers, switches y servidores de VPN.

Creación de Políticas de Escaneo

Tabla 32

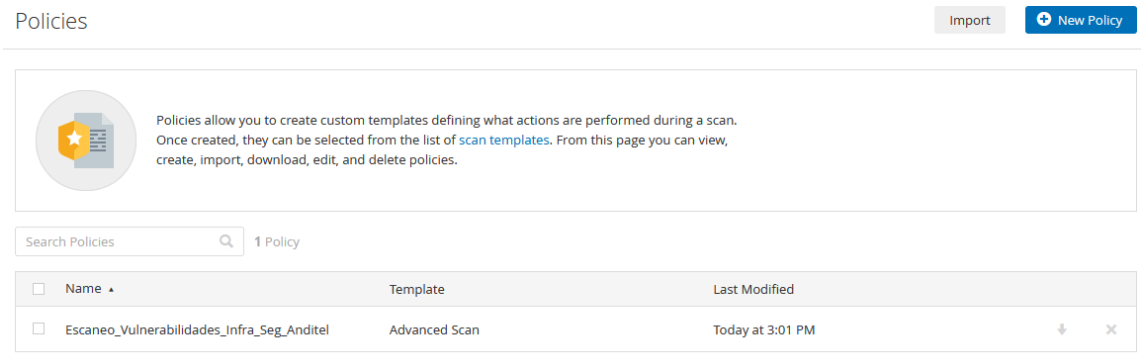
Definición de Políticas de Escaneo

Sección	Configuración	Descripción
Plugins	Habilitados por defecto.	Al dejar habilitados todos los pugins por defecto obtendremos resultados más avanzados en el escaneo de vulnerabilidades
Discovery	Ping: TCP SYN Ping + ARP Ping	Descubre Host evitando filtros ICMP
Port Scanning	• Port Scanniing: SYN Scan Service Discovery: Configuración por defecto.	Sigiloso y rápido (como Scapym usado en fases anteriores.)
Assesment	• Servicios:SSH, Telnet, FTP, HTTP/HTTPS, SNMP	Servicios expuestos en los activos previamente identificados.
Advance	• Timeouts: 5 segundos	Evita falsos positivos en dispositivos legacy

Nota. La presenta tabla describe las Políticas de Escaneo

Figura 51

Creación de Política de Escaneo en Nessus.



Nota. Configuración de política de escaneo.

Vulnerabilidades Encontradas

Tras la finalización del escaneo de vulnerabilidades sobre la infraestructura de seguridad de la empresa proveedora de servicios de Internet (ISP), mediante la herramienta Nessus, se genera los informes detallados de los hallazgos encontrados.

Resumen de Riesgo

Tabla 33

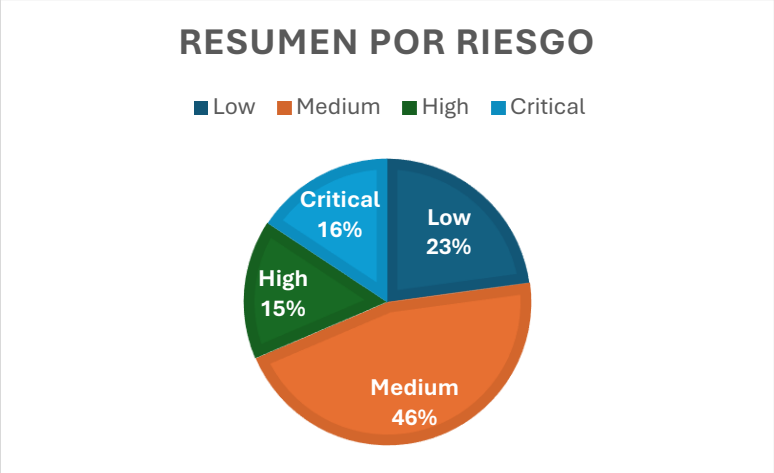
Vulnerabilidades Encontradas

Resumen por Riesgo	
Nivel de riesgo	**Cantidad**
Low	16
Medium	32
High	11
Critical	11

Nota. Vulnerabilidades encontradas

Figura 52

Resumen de Vulnerabilidades de Acuerdo con su Riesgo.



Nota. Resumen de vulnerabilidades.

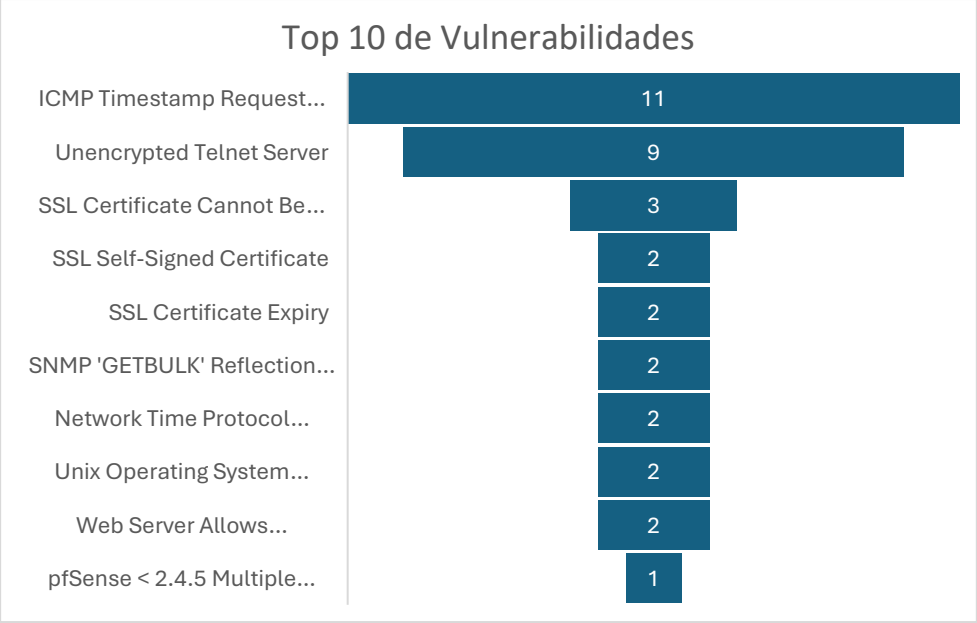
De acuerdo con los resultados obtenidos del análisis, se encontró que el 16% de las vulnerabilidades encontradas son críticas, 15 % son vulnerabilidades altas, 46% medias y 23 bajas.

Top 10 de Vulnerabilidades Encontradas

La siguiente grafica representa el top 10 de las vulnerabilidades más recurrentes en los activos escaneados.

Figura 53

Top de Vulnerabilidades Encontradas



Nota. vulnerabilidades encontradas.

La vulnerabilidad (ICMP Timestamp Request Remote Date Disclosure), se encuentra presente en todos los activos escaneados. Esta vulnerabilidad fue descubierta con el Pugin 10114 de Nessus.

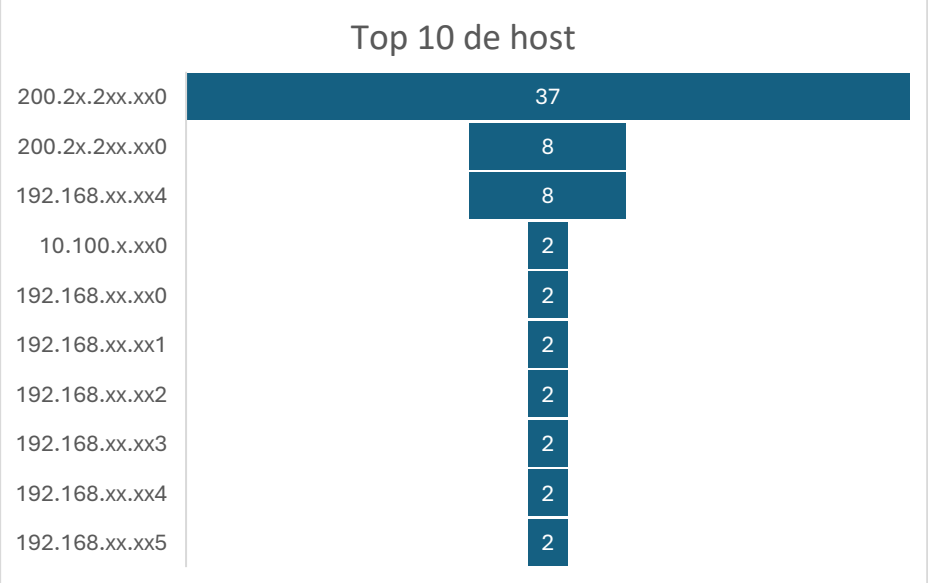
Todos los activos escaneados pueden responder a una marca de tiempo ICMP, con esta vulnerabilidad, se permite al atacante conocer la fecha establecida en el activo objetivo y ayudar a un actor externo/interno no autenticado a evadir protocolos de autenticación basado en el tiempo. Según el CVSS v3.1, esta vulnerabilidad tiene un *Base Score* de 2.1 (Severidad: Baja), clasificándose como de riesgo tolerable para la infraestructura (ICMP Timestamp Request Remote Date Disclosure, s.f.).

La vulnerabilidad (Unencrypted Telnet Server), se encuentra presente en 9 activos escaneados, siendo la segunda vulnerabilidad más recurrente en los activos. Esta vulnerabilidad fue hallada con el plugin 42263 de Nessus. El protocolo Telnet, se considera inseguro al transmitir información en texto plano, esto puede ser explotado por un atacante externo/interno a interceptar la información para obtener credenciales y modificar el tráfico intercambiado entre cliente y servidor. Esta vulnerabilidad cuenta con una base score: 5.8 y se considera de severidad Media.

Top 10 de Host con más Vulnerabilidades

Figura 54

Top de Activos con más Vulnerabilidades



Nota. Activos con más vulnerabilidades.

El top 3 de los activos con más vulnerabilidades encontradas, corresponde a activos en puesto 1 y 2 a Firewall internos. Estos activos cuentan con un sistema operativo sin soporte actual con el fabricante. El activo número tres en la lista, corresponde al switch de distribución.

Documentar Hallazgos Obtenidos del Análisis de Vulnerabilidades

Vulnerabilidades por activo.

Tabla 34

Criticidad de Vulnerabilidades por Activo

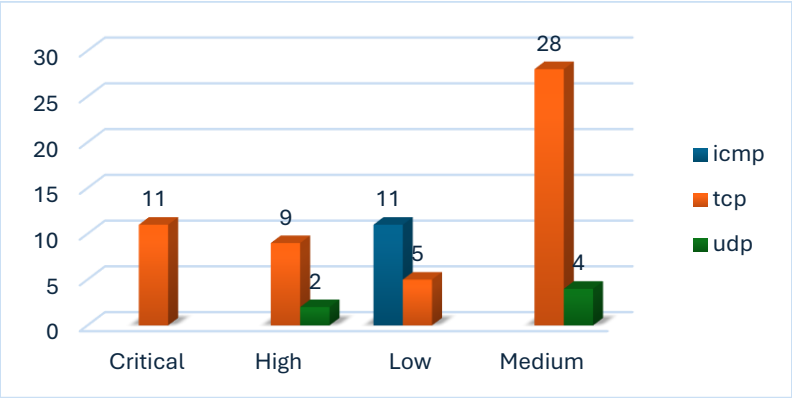
Activo	Critica	Alta	Baja	Media	Total, general
10.100.xx.xx0			1	1	2
192.168.xx.xx0			1	1	2
192.168.xxx.xx1			1	1	2
192.168.xx.xx2			1	1	2
192.168.xxx.xx3			1	1	2
192.168.xx.xx4			1	1	2
192.168.xx.xx5			1	1	2
192.168.xx.xx6				2	2
192.168.x.xx4		3	1	4	8
200.25.xxx.xx0	10	8	5	14	37
200.25.2xx.xx4			1		1
200.25.2xx.xx0	1		2	5	8
Total, general	**11**	**11**	**16**	**32**	**70**

Nota. La presente tabla resume la criticidad de vulnerabilidades por activo.

Vulnerabilidades por Protocolo

Figura 55

Vulnerabilidades Encontradas por Protocolo.

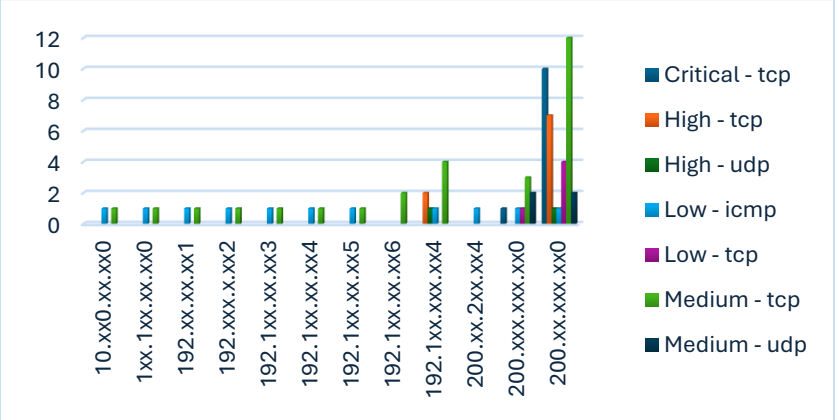


Nota. Vulnerabilidades encontradas.

De acuerdo con la anterior imagen, el protocolo TCP tiene el mayor número de vulnerabilidades descubiertas en todos los riegos establecidos (Critico, Alto, Medio y Bajo). El protocolo ICMP tiene el mayor número de vulnerabilidades descubiertas en la categoría de riesgo Bajo. El protocolo UDP tiene un menor números de vulnerabilidades descubiertas, sin embargo, cuenta con 2 vulnerabilidades en Riesgo Alto y Medio.

Figura 56

Activos con Mayores Vulnerabilidades Descubiertas.



Nota. Dispositivos con mayores vulnerabilidades descubiertas.

La anterior imagen, representa los activos más afectados por vulnerabilidades descubiertas en los protocolos TCP, ICMP y UDP. Los activos con mayor riesgo corresponden a Firewall Internos y Switch LAN.

Resumen de Vulnerabilidades Encontradas

Tabla 35

Resumen de Vulnerabilidades CVE

IP	Activo	Riesgo	Score	CVE	Descripción
2xx.2x.2xx.xx0	Firewall Interno	Critical	9,8	CVE-2014-8176	Versión instalada de pfSense es anterior a la 2.2.3. afectación por múltiples vulnerabilidades.
2xx.2x.2xx.xx0	Firewall Interno	Critical	9	CVE-2015-3194	Versión instalada de pfSense es anterior a la 2.2.6. afectación por múltiples vulnerabilidades.

200.25.224.130	Firewall Interno	Critical	9,8	CVE-2015-3197	Versión instalada de pfSense es anterior a la 2.2.3. afectación por múltiples vulnerabilidades.
2xx.2x.2xx.xx0	Firewall Interno	Critical	9,8	CVE-2016-10009	Versión de pfSense desactualizada,afectada por múltiples vulnerabilidades.
2xx.2x.2xx.xx0	Firewall Interno	Critical	9,8	CVE-2016-10195	Versión de pfSense desactualizada,afectada por múltiples vulnerabilidades.
2xx.2x.2xx.xx0	Firewall Interno	Critical	9,8	CVE-2017-12837	Versión de pfSense desactualizada,afectada por múltiples vulnerabilidades.
2xx.2x.2xx.xx0	Firewall Interno	Critical	9,8	CVE-2017-5715	Versión instalada de pfSense es anterior a la 2.4.3 afectación por múltiples vulnerabilidades.
200.25.224.130	Firewall Interno	Critical	9,8	CVE-2019-12462	Versión desactualizada de pfSense. Por lo tanto, se ve afectada por múltiples vulnerabilidades, entre ellas; (CVE-2019-12949), (CVE-2019-16914), (CVE-2019-16915)
19x.xx.x.xx4	Switch Distribucion	High		CVE-2014-4705	Firmware obsoleto afectada por una vulnerabilidad de denegación de servicio.
19x.xx.x.xx4	Switch Distribucion	High		CVE-2014-4190	Ffirmware afectada por una vulnerabilidad de denegación de servicio debido a múltiples problemas de desbordamiento de pila.
19x.xx.x.xx4	Switch Distribucion	High		CVE-2007-5846	Es posible deshabilitar el demonio SNMP remoto enviando una solicitud GETBULK 'max-repetitions'.
2xx.2x.2xx.xx	Firewall Interno	High	7,1	CVE-2015-2294	Versión instalada de pfSense es anterior a la 2.2.1. afectación por múltiples vulnerabilidades.
2xx.2x.2xx.xx0	Firewall Interno	High	7,5	CVE-2015-3152	Versión instalada de pfSense es anterior a la 2.2.4. afectación por múltiples vulnerabilidades.
2xx.2x.2xx.xx0	Firewall Interno	High	7,8	CVE-2014-2653	Versión instalada de pfSense es anterior a la 2.2.5. afectación por múltiples vulnerabilidades.

2xx.2x.2xx.xx0	Firewall Interno	High	7,8	CVE-2016-1886	Versión instalada de pfSense es anterior a la 2.3.1. afectación por múltiples vulnerabilidades.
2xx.2x.2xx.xx0	Firewall Interno	High	8,6	CVE-2013-7456	Versión instalada de pfSense es anterior a la 2.3.1-p5 afectación por múltiples vulnerabilidades.
2xx.2x.2xx.xx0	Firewall Interno	High	7,5	CVE-2018-20798	Versión desactualizada de pfSense. Afectado por múltiples vulnerabilidades, entre ellas; (CVE-2018-20799), (CVE-2018-20798)
2xx.2x.2xx.xx0		High	7,5	CVE-2016-7434	El servidor NTP remoto se ve afectado por una vulnerabilidad de denegación de servicio debido a la validación incorrecta de las consultas mrulist.
19x.xx.x.xx4	Switch Distribucion	Medium		CVE-2014-4707	El host remoto es un switch Huawei con una versión de firmware afectada por múltiples vulnerabilidades debido a fallos en los menús de arranque y BootROM.
19x.xx.x.xx4	Switch Distribucion	Medium		CVE-2014-5394	El conmutador remoto Huawei se ve afectado por una vulnerabilidad de divulgación de información.
2xx.2x.2xx.xx0	Firewall Interno	Medium	6,5	CVE-2017-1086	Versión instalada de pfSense es anterior a la 2.4.2 afectación por múltiples vulnerabilidades.
2xx.2x.2xx.xx0	Firewall Interno	Medium		CVE-2008-4309	El demonio SNMP remoto responde con una gran cantidad de datos a una solicitud 'GETBULK' con un valor de 'max-repetitions' mayor de lo normal.
2xx.2x.2xx.xx0	Firewall Interno	Medium	5,9	CVE-2023-48795	El servidor SSH remoto es vulnerable a una vulnerabilidad de truncamiento de prefijo de intermediario conocida como Terrapin.
2xx.2x.2xx.xx0	Firewall Interno	Medium	5,9	CVE-2016-2107	El host remoto se ve afectado por una vulnerabilidad de divulgación de información de intermediario (MitM) debido a un error en la implementación de conjuntos de cifrado que utilizan AES en modo CBC con HMAC-SHA1 o HMAC-SHA256.
2xx.2x.2xx.xx0	Firewall Interno	Medium	6,1	CVE-2020-11022	Según la versión autodeclarada en el script, la versión de jQuery alojada en el servidor web remoto es superior o igual a la 1.2 y anterior a la 3.5.0. Por lo tanto, se ve afectada por múltiples vulnerabilidades de scripts entre sitios.

200.25.225.130	Firewall Interno	Medium		CVE-2008-4309	El demonio SNMP remoto responde con una gran cantidad de datos a una solicitud 'GETBULK' con un valor de 'max-repetitions' mayor de lo normal.
19x.xx.x.xx0	Switch Acceso	Low		CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.
19x.xx.x.xx 0	Switch Acceso	Low		CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.
19x.xx.x.xx 1	Switch Acceso	Low		CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.
19x.xx.x.xx 2	Switch Acceso	Low		CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.
19x.xx.x.xx 3	Switch Acceso	Low		CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.
19x.xx.x. xx 4	Switch Acceso	Low		CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.
19x.xx.x.xx5	Router Core	Low		CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.
19x.xx.x.xx4	Router Core	Low		CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.
200.2x.2xx.xx0	Firewall Interno	Low		CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.
200.2x.2xx.xx0	Firewall Interno	Low	3, 7	CVE-2008-5161	El servidor SSH está configurado para admitir el cifrado CBC (Cifrado por Bloques).
200.2x.2xx.xx4	Servidor VPN	Low		CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.

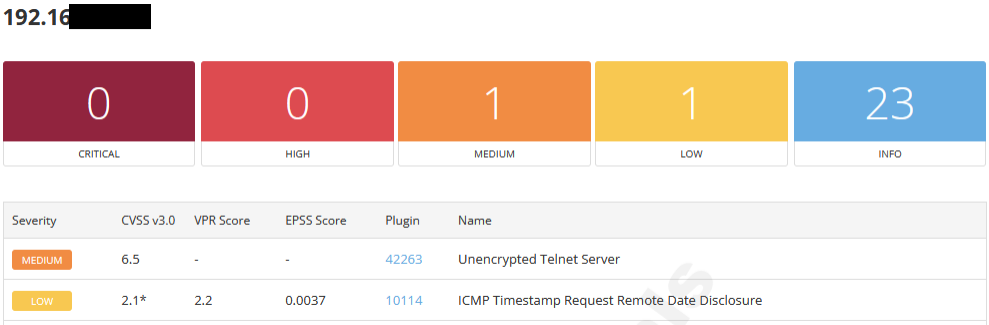
200.2x.2xx.xx0	Firewall Interno	Low	CVE-1999-0524	El host remoto responde a una solicitud de marca de tiempo ICMP.

Nota. La presente tabla resume las vulnerabilidades CVE.

Detalle de Vulnerabilidades por Activo

Figura 57

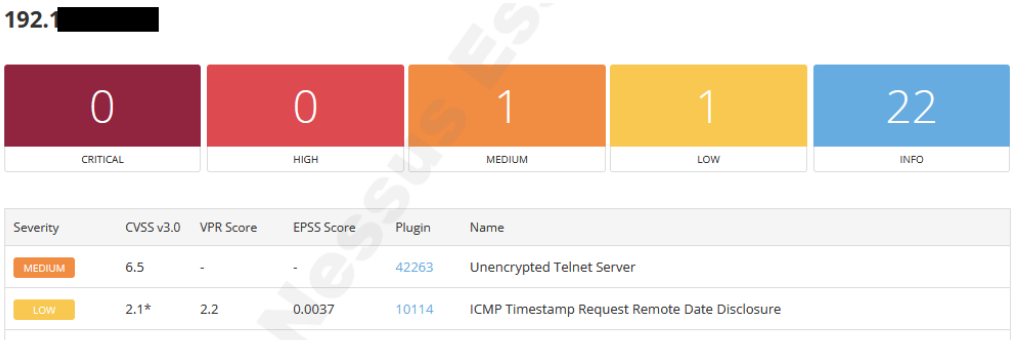
Escaneo 192.16x.xx.xx



Nota. Captura de pantalla de resultado generado por Nessus.

Figura 58

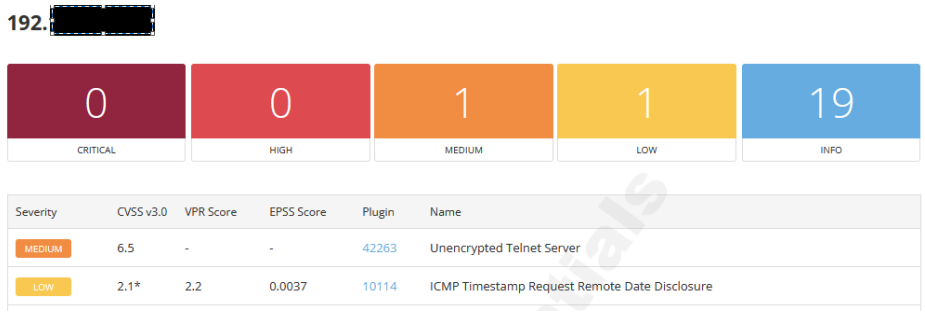
Escaneo 192.16x.xx.xxx



Nota. Captura de pantalla de resultado generado por Nessus.

Figura 59

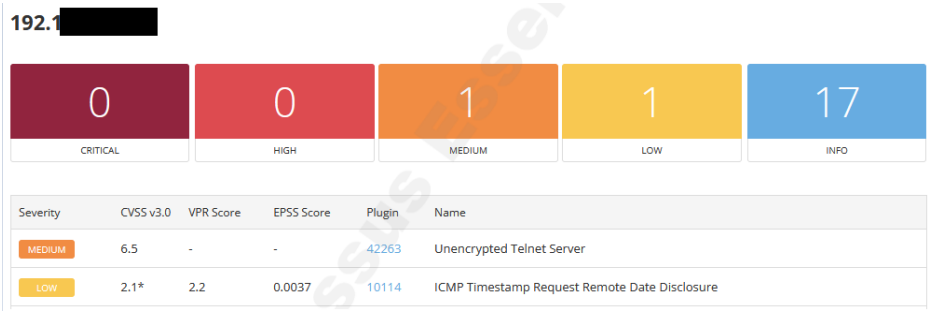
Escaneo 192.16x.xxx.xxx



Nota. Captura de pantalla de resultado generado por Nessus.

Figura 60

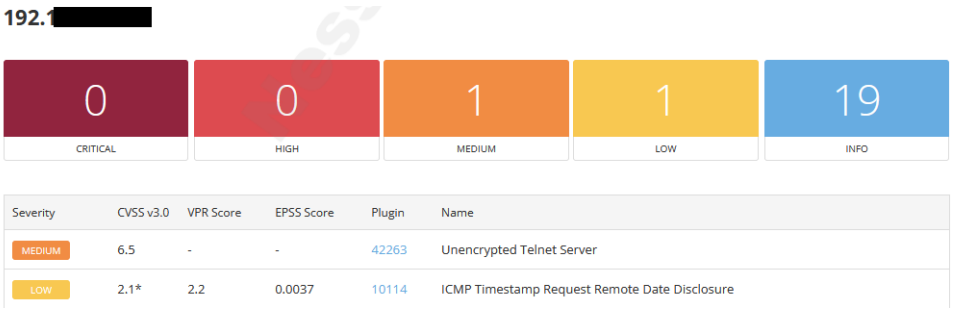
Escaneo 192.xx.xx.xxx



Nota. Captura de pantalla de resultado generado por Nessus.

Figura 61

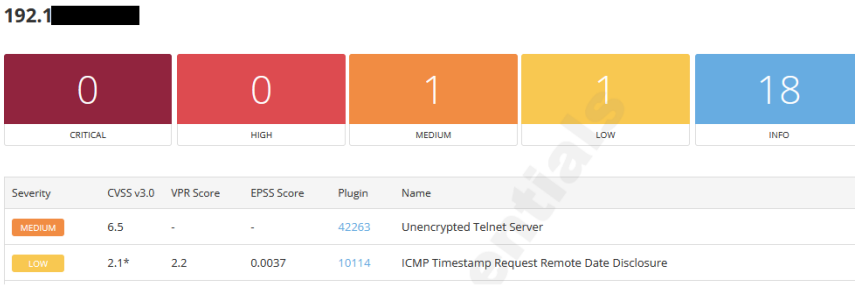
Escaneo 192.16x.xx.xxx



Nota. Captura de pantalla de resultado generado por Nessus.

Figura 62

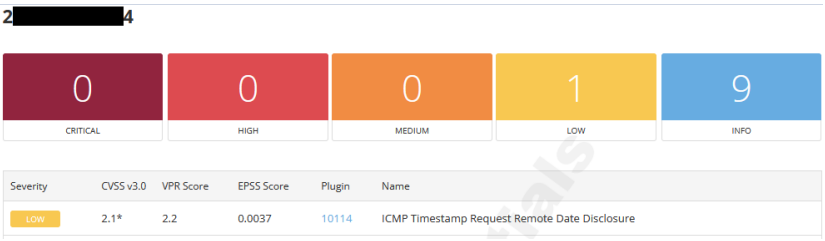
Escaneo 192.16x.xx.xx



Nota. Captura de pantalla de resultado generado por Nessus.

Figura 63

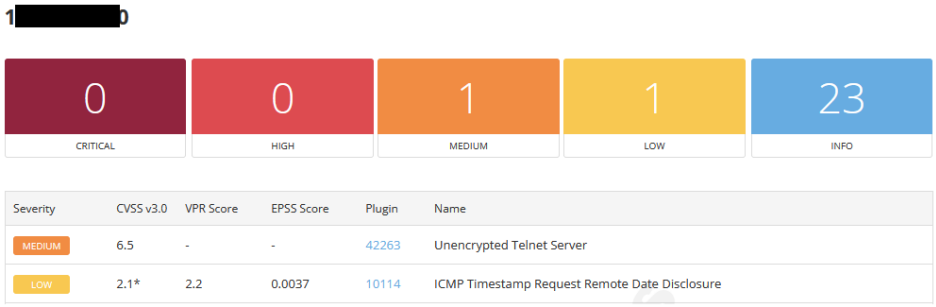
Escaneo 2x0.2xx.xx.xxx



Nota. Captura de pantalla de resultado generado por Nessus.

Figura 64

Escaneo 1xx.xx.xx.xxx



Nota. Captura de pantalla de resultado generado por Nessus.

Figura 65

ICMP Timestamp Request Remote Date Disclosure

Vulnerabilidad: ICMP Timestamp Request Remote Date Disclosure	CVE: CVE-1999-0524
Impacto: Bajo	Riesgo: Tolerable

Activo afectado:

10.168.xx.xx0

10.168.xx.xx1

10.168.xx.xx2

10.168.xx.xx3

10.168.xx.xx4

10.168.xx.xx5

10.100.xx.xx0

200.2xx.2xx.xx0

200.2xx.2xx.xx0

192.168.xx.xx5

192.168.xx.xx6

Descripción de vulnerabilidad: El host remoto responde a una solicitud de marca de tiempo ICMP. Esto permite a un atacante conocer la fecha establecida en la máquina objetivo, lo que podría ayudar a un atacante remoto no autenticado a evadir los protocolos de autenticación basados en tiempo.
Impacto vulnerabilidad: Un atacante externo/interno puede obtener información adicional del equipo como por ejemplo fecha y hora de configuración, lo cual sería información útil para ataques más avanzados.
Mitigación: Crear reglas en el firewall para solicitudes de marca de tiempo ICMP (13) y las respuestas de marca de tiempo ICMP salientes (14).

Nota. *Resultados obtenidos de Nessus.*

Tabla 36

Unencrypted Telnet Server

Vulnerabilidad: Unencrypted Telnet Server	CVE: N/A
Impacto: Medio	Riesgo: Medio
Activo afectado: 10.168.xx.xx0 10.168.xx.xx1 10.168.xx.xx2 10.168.xx.xx3 10.168.xx.xx4 10.168.xx.xx5 10.100.xx.xx0 192.168.xx.xx5 192.168.xx.xx6	
Descripción de vulnerabilidad: El host remoto ejecuta un servidor Telnet a través de un canal sin cifrar. No se recomienda usar Telnet a través de un canal sin cifrar, ya que los nombres de usuario, las contraseñas y los comandos se transfieren en texto plano. Esto permite que un atacante remoto, intermediario, espíe una sesión Telnet para obtener credenciales u otra información confidencial, y modificar el tráfico intercambiado entre un cliente y un servidor.	

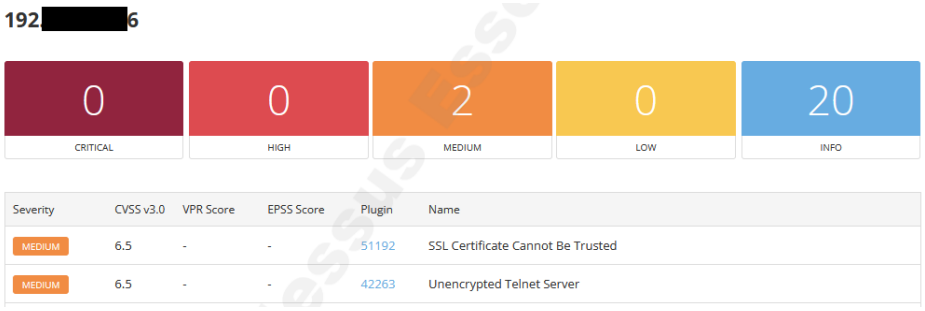
Se prefiere SSH a Telnet, ya que protege las credenciales de escuchas no autorizadas y puede tunelizar flujos de datos adicionales, como una sesión X11.
Impacto vulnerabilidad: Un atacante externo/interno puede capturar usuario-contraseña e inyectar comandos en los equipos que afectan la configuración y funcionamiento de este.
Mitigación: Deshabilite el servicio Telnet y utilice SSH en su lugar.

Nota. *Resultados obtenidos de Nessus*

.

Figura 66

Escaneo 192.1xx.xxx.xx



Nota. Captura de pantalla de resultado generado por Nessus.

Tabla 37

SSL Certificate Cannot Be Trusted

Vulnerabilidad: SSL Certificate Cannot Be Trusted	CVE: N/A
Impacto: Medio	Riesgo: Medio
Activo afectado: 192.168.x.xx6 200.xx.xx4.xx0	

200.xx.xx5.xx0

Descripción de vulnerabilidad: No se puede confiar en el certificado X.509 del servidor. Esta situación puede ocurrir de tres maneras diferentes, en las que la cadena de confianza puede romperse, como se indica a continuación:

- Primero, el extremo superior de la cadena de certificados enviados por el servidor podría no provenir de una autoridad de certificación pública conocida. Esto puede ocurrir cuando el extremo superior de la cadena es un certificado auto firmado no reconocido, o cuando faltan certificados intermedios que conectarían el extremo superior de la cadena de certificados con una autoridad de certificación pública conocida.

- Segundo, la cadena de certificados puede contener un certificado no válido en el momento del escaneo. Esto puede ocurrir cuando el escaneo se realiza antes de una de las fechas "notBefore" del certificado o después de una de las fechas "notAfter".

- Tercero, la cadena de certificados puede contener una firma que no coincide con la información del certificado o que no se pudo verificar. Las firmas incorrectas se pueden corregir solicitando que el emisor vuelva a firmar el certificado con la firma incorrecta. Las firmas que no se pudieron verificar se deben a que el emisor del certificado utilizó un algoritmo de firma que Nessus no admite o no reconoce.

Si el host remoto es un host público en producción, cualquier interrupción en la cadena dificulta que los usuarios verifiquen la autenticidad e identidad del servidor web. Esto podría facilitar la ejecución de ataques de intermediario contra el host remoto.

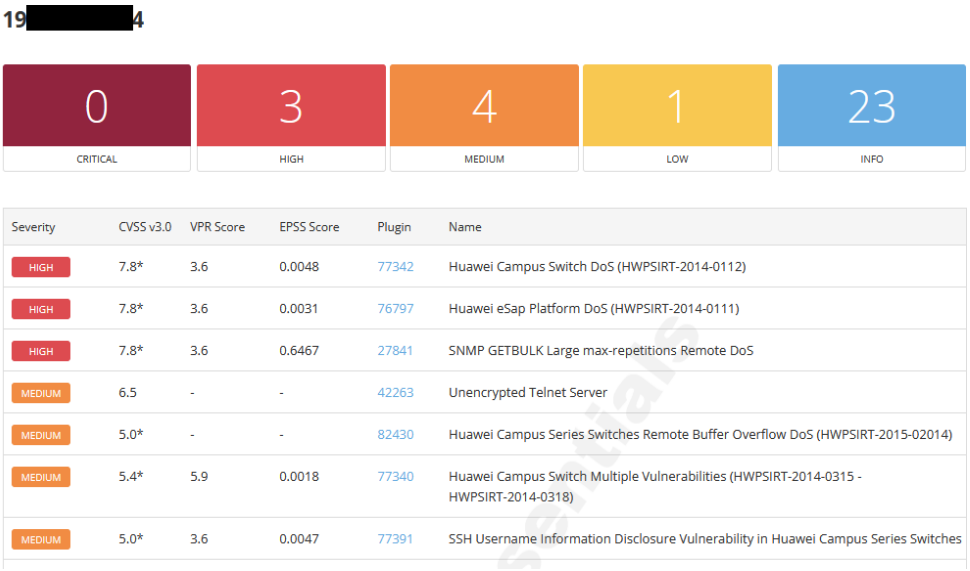
Impacto vulnerabilidad: Un atacante externo/interno puede capturar el tráfico y aprovechar la falta de verificación de certificados. Usuarios pueden ignorar las advertencias y ser víctimas de ataque tipo man-in-the-middle (MITM).

Mitigación: Comprar o generar un certificado SSL adecuado para este servicio.

Nota. *Resultados obtenidos de Nessus.*

Figura 67

Escaneo 192.1xx.xx.xx



Nota. Captura de pantalla de resultado generado por Nessus.

Tabla 38

Huawei Campus Switch DoS (HWPSIRT-2014-0112)

Vulnerabilidad: Huawei Campus Switch DoS (HWPSIRT-2014-0112)	CVE: CVE-2014-4190
Impacto: Alto	Riesgo: Alto
Activo afectado: 192.168.xx.xx4	
Descripción de vulnerabilidad: El host remoto es un dispositivo Huawei con una versión de firmware afectada por una vulnerabilidad de denegación de servicio debido a múltiples problemas de desbordamiento de pila. Un ataque remoto no autenticado podría explotar esta vulnerabilidad enviando paquetes malformados para provocar el reinicio del dispositivo.	

Impacto vulnerabilidad: El dispositivo puede recibir paquetes diseñados (maliciosos), que pueden reiniciar el equipo o volverse inestable, interrumpiendo la conectividad de toda la red o segmento.

Mitigación: Comunicarse con el proveedor y aplicar actualización pertinente.

Nota. *Resultados obtenidos de Nessus.*

Tabla 39

Huawei eSap Platform DoS (HWPSIRT-2014-0111).

Vulnerabilidad: Huawei eSap Platform DoS (HWPSIRT-2014-0111)	CVE: CVE-2014-4705
Impacto: Alto	Riesgo: Bajo

Activo afectado: 192.168.xx.xx4

Descripción de vulnerabilidad: El host remoto es un dispositivo Huawei con una versión de firmware afectada por una vulnerabilidad de denegación de servicio. El problema se debe a una vulnerabilidad de desbordamiento de pila en el firmware. Un atacante remoto no autenticado podría explotar esta vulnerabilidad enviando paquetes malformados para provocar un consumo excesivo de memoria o el reinicio del dispositivo.
Impacto vulnerabilidad: Afecta principalmente los servicios de red que dependen de eSAP que puede causar; falla en autenticación de usuario, caida temporal del servicio de gestión o acceso a la red, reinició o inestabilidad del switch afectado. * empresa proveedora de servicios de Internet (ISP), no implementa eSAP.

Mitigación: Comunicarse con el proveedor y aplicar actualización pertinente.

Nota. *Resultados obtenidos de Nessus.*

Tabla 40

SNMP Getbulk Large Max-Repetitions Remote DoS

	CVE: CVE-2007-5846

Vulnerabilidad: SNMP GETBULK Large max-repetitions Remote DoS.	
Impacto: Alto	Riesgo: Alta

Activo afectado: 192.168.xx.xx4

Descripción de vulnerabilidad: Es posible deshabilitar el demonio SNMP remoto enviando una solicitud GETBULK con un valor alto para 'max-repetitions'. Un atacante remoto podría aprovechar este problema para provocar que el demonio consuma demasiada memoria y CPU en el sistema afectado mientras intenta procesar la solicitud sin éxito, denegando así el servicio a usuarios legítimos.

Impacto vulnerabilidad: un atacante o un error en la configuración de monitoreo del equipo, puede enviar solicitudes SNMP GETBULK maliciosas ejemplo: max-repetitions = 10000, generando una excesiva carga al procesador del switch al intentar recuperar grandes cantidades de datos SNMP.

Mitigación: Deshabilite el servicio SNMP en el host remoto si no lo utiliza. De lo contrario, actualice a la versión 5.4.1 o posterior si utiliza Net-SNMP.

Nota. *Resultados obtenidos de Nessus*

Tabla 41

Huawei Campus Series Switches Remote Buffer Overflow DoS (HWPSIRT-2015-02014)

Vulnerabilidad: Huawei Campus Series Switches Remote Buffer Overflow DoS (HWPSIRT-2015-02014)	CVE: N/A
Impacto: Medio	Riesgo: Alto

Activo afectado: 192.168.xx.xx4

Descripción de vulnerabilidad: El conmutador remoto Huawei se ve afectado por una vulnerabilidad de denegación de servicio debido a la validación incorrecta de la entrada proporcionada por el

usuario a la función de procesamiento del servicio. Un atacante remoto, utilizando un nombre de usuario especialmente diseñado, puede causar una violación de acceso a la matriz, lo que provoca el reinicio del dispositivo.

Impacto vulnerabilidad: un atacante remoto con acceso a la red puede enviar paquetes especialmente diseñados a travez de la red para provocar desbordamiento de buffer afectando gravemente la disponibilidad del equipo como: reinicio inesperado del equipo, perdida de conectividad temporal o permanentemente, interrupcion del forwarding de paquetes o autentocación de usuarios, falla en los servicios de gestión (CLI, SNMP,web UI)

Mitigación: Comunicarse con el proveedor y aplicar actualización pertinente.

Nota. *Resultados obtenidos de Nessus*

Tabla 42

Huawei Campus Switch Multiple Vulnerabilities (HWPSIRT-2014-0315 - HWPSIRT-2014-0318)

Vulnerabilidad: Huawei Campus Switch Multiple Vulnerabilities (HWPSIRT-2014-0315 - HWPSIRT-2014-0318)	CVE: CVE-2014-4707
Impacto: Medio	Riesgo: Alto

Activo afectado: 192.168.xx.xx4

Descripción de vulnerabilidad: El host remoto es un switch Huawei con una versión de firmware afectada por múltiples vulnerabilidades debido a fallos en los menús de arranque y BootROM. Un atacante remoto no autenticado podría aprovechar estas vulnerabilidades para tomar el control del dispositivo.

Impacto vulnerabilidad: Critico debido a multiples variantes DoS (0315,0316,0317)
Mitigación: Comunicarse con el proveedor y aplicar actualización pertinente.

Nota. *Resultados obtenidos de Nessus*

Tabla 43

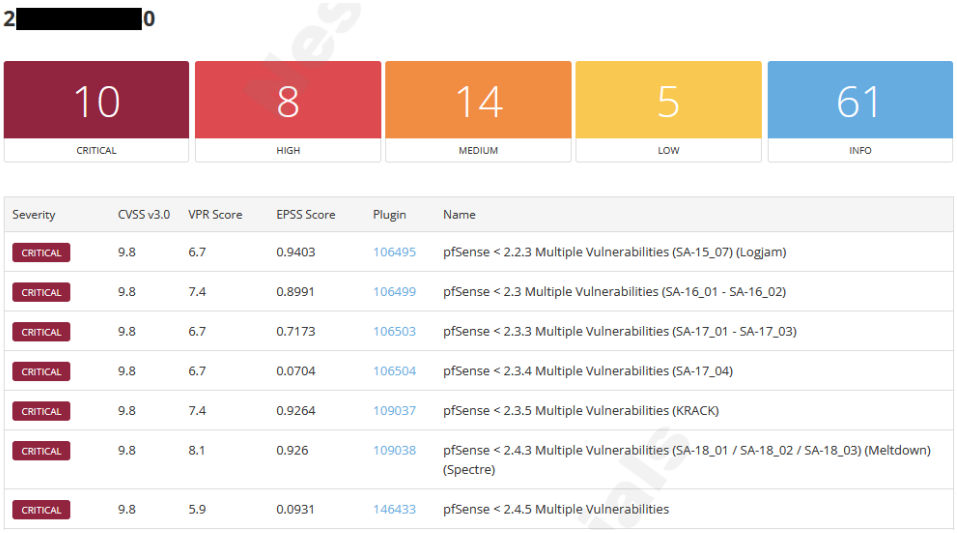
SSH Username Information Disclosure Vulnerability in Huawei Campus Series Switches

Vulnerabilidad: SSH Username Information Disclosure Vulnerability in Huawei Campus Series Switches	CVE: CVE-2014-5394
Impacto: Medio	Riesgo: Medio
Activo afectado: 192.168.xx.xx4	
Descripción de vulnerabilidad: El conmutador remoto Huawei se ve afectado por una vulnerabilidad de divulgación de información. Al examinar la respuesta de su servidor SSH al intentar iniciar sesión, un atacante remoto puede verificar si existe un nombre de usuario desconocido en el dispositivo.	
Impacto vulnerabilidad: Se divulga información sensible (nombres de usuario del sistema)	
Mitigación: Aplicar el parche de firmware apropiado.	

Nota. *Resultados obtenidos de Nessus*

Figura 68

Escaneo 2x0.2xx.xx.xx



Nota. Captura de pantalla de resultado generado por Nessus.

Tabla 44

PfSense < 2.2.3 Multiple Vulnerabilities (SA-15_07) (Logjam)

Vulnerabilidad: pfSense < 2.2.3 Multiple Vulnerabilities (SA-15_07) (Logjam)	CVE: CVE-2014-8176
Impacto: Critico	Riesgo: Critico
Activo afectado: 200.xx.xxx.xx0	
Descripción de vulnerabilidad: Según el número de versión declarado, la instalación remota de pfSense es anterior a la 2.2.3. Por lo tanto, se ve afectada por múltiples vulnerabilidades, como se indica en los avisos de los proveedores mencionados.	

Impacto vulnerabilidad: Un atacante puede espiar o descifrar sesiones TLS cifradas débilmente, no se afecta la disponibilidad.
Mitigación: Actualizar a la versión 2.2.3 o posterior de pfSense.
Vulnerabilidad: pfSense < 2.3 Multiple Vulnerabilities (SA-16_01 - SA-16_02) — CVE: CVE-2015-3197
Impacto: Critico — Riesgo: Critico
Activo afectado: 200.xx.xxx.xx0
Descripción de vulnerabilidad: Según el número de versión declarado, la instalación remota de pfSense es anterior a la 2.3. Por lo tanto, presenta múltiples vulnerabilidades.
Impacto vulnerabilidad: Potencialmente crtico en entornos con múltiples administradores o acceso remoto habilitado.
Mitigación: Actualizar a la versión 2.2.3 o posterior de pfSense, restringir el acceso a la interfaz de administración.

Nota. Resultados obtenidos de Nessus

Tabla 45

PfSense < 2.3.3 Multiple Vulnerabilities (SA-17_01 - SA-17_03)

Vulnerabilidad: pfSense < 2.3.3 Multiple Vulnerabilities (SA-17_01 - SA-17_03) — CVE: CVE-2016-10009
Impacto: Critico — Riesgo: Critico
Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: Según el número de versión informado por el usuario, la instalación remota de pfSense se ve afectada por múltiples vulnerabilidades, como se indica en los avisos de los proveedores mencionados.

Impacto vulnerabilidad: Potencialmente crtico en entornos con múltiples administradores o acceso remoto habilitado.

Mitigación: Actualizar a la versión 2.2.3 o posterior de pfSense.

Nota. *Resultados obtenidos de Nessus*

Tabla 46

PfSense < 2.3.5 Multiple Vulnerabilities (KRACK)

Vulnerabilidad: pfSense < 2.3.5 Multiple Vulnerabilities (KRACK)	CVE: CVE-2017-12837
Impacto: Critico	Riesgo: Bajo

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: Según el número de versión informado por el usuario, la instalación remota de pfSense se ve afectada por múltiples vulnerabilidades, como se indica en los avisos de los proveedores mencionados.

Impacto vulnerabilidad: puede descifrar el tráfico WI-FI, capturar credenciales, sesiones o datos sensibles, aplica si el firewall cuenta con interfaces WI-FI. No aplica para firewall para la empresa proveedora de servicios de Internet (ISP)

Mitigación: Actualizar a la versión 2.2.3 o posterior de pfSense, Evitar el uso de Wi-Fi en el firewall si no es estrictamente necesario.

Nota. *Resultados obtenidos de Nessus*

Tabla 47

PfSense < 2.4.3 Multiple Vulnerabilities (SA-18_01 / SA-18_02 / SA-18_03) (Meltdown) (Spectre)

Vulnerabilidad: pfSense < 2.4.3 Multiple Vulnerabilities (SA-18_01 / SA-18_02 / SA-18_03) (Meltdown) (Spectre)	CVE: CVE-2017-5715
Impacto: Critico	Riesgo: Medio

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: Según el número de versión auto informado, la instalación remota de pfSense es anterior a la 2.4.3. Por lo tanto, se ve afectada por múltiples vulnerabilidades, como se indica en los avisos del proveedor mencionado.

Impacto vulnerabilidad: Potencialmente peligroso solo sí el atacante tiene acceso fisico por shell remoto al equipo. Afecta especialmente a sistemas virtualizados, donde una VM maliciosa podría leer la memoria del host u otras VMs.

Mitigación: Actualizar a la versión 2.2.3 o posterior de pfSense, restringir el acceso Shell/local al sistema solo a administradores de confianza. Deshabilitar acceso por SSH o consola remota si no es indispensable.

Nota. *Resultados obtenidos de Nessus.*

Tabla 48

PfSense < 2.4.5 Multiple Vulnerabilities

Vulnerabilidad: pfSense < 2.4.5 Multiple Vulnerabilities	CVE: CVE-2019-12462
Impacto: Critico	Riesgo: Alto

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: Según el número de versión autodeclarado, la instalación remota de pfSense corresponde a la versión 2.4.5. Por lo tanto, se ve afectada por múltiples vulnerabilidades, entre ellas:
- En pfSense 2.4.4-p2 y 2.4.4-p3, si se logra engañar a un administrador autenticado para que haga clic en un botón de una página de phishing, un atacante puede usar XSS para subir código ejecutable arbitrario a un servidor mediante diag_command.php y rrd_fetch_json.php (parámetro timePeriod). Posteriormente, el atacante remoto puede ejecutar cualquier comando con privilegios de root en ese servidor. (CVE-2019-12949)
- Se descubrió un problema de XSS en pfSense hasta la versión 2.4.4-p3. En services_captiveportal_mac.php, los parámetros username y delmac se muestran sin desinfectar. (CVE-2019-16914)
Se detectó un problema en pfSense hasta la versión 2.4.4-p3. widgets/widgets/picture.widget.php utiliza el parámetro widgetkey directamente sin corrección (por ejemplo, una llamada a basename) para una ruta a file_get_contents o file_put_contents. (CVE-2019-16915)
Tenga en cuenta que Nessus no ha realizado pruebas para detectar estos problemas, sino que se ha basado únicamente en el número de versión informado por la propia aplicación.
- En pfSense 2.4.4-p2 y 2.4.4-p3, si es posible engañar a un administrador autenticado para que haga clic en un botón de una página de phishing, un atacante puede usar XSS para subir código ejecutable arbitrario a un servidor mediante diag_command.php y rrd_fetch_json.php (parámetro timePeriod). Posteriormente, el atacante remoto puede ejecutar cualquier comando con privilegios de root en ese servidor. (CVE-2019-12949)
Impacto vulnerabilidad: Un atacante puede ejecutar comandos ue expongan datos sensibles, coo archivos de configuración, claves o contraseñas. El atacante puede alterar archivos del sistema, instalar paquetes maliciosos, modificar reglas o configuraciones. Puede causar interrupciones o inutilizar el firewall con código malicioso.
Mitigación: Actualizar a la versión 2.2.3 o posterior de pfSense, Asegurar que pfSense use solo repositorios oficiales y conexiones HTTPS verificadas, Configurar DNS confiables o usar DNS sobre TLS/HTTPS si es posible, No permitir tráfico saliente innecesario desde el firewall a Internet sin control. Evitar el uso de repositorios de paquetes personalizados.

Nota. *Resultados obtenidos de Nessus*

Tabla 49

PfSense < 2.2.6 Multiple Vulnerabilities (SA-15_09 / SA-15_10 / SA-15_11)

Vulnerabilidad: pfSense < 2.2.6 Multiple Vulnerabilities (SA-15_09 / SA-15_10 / SA-15_11)	CVE: CVE-2015-3194
Impacto: Critico	Riesgo: Critica
Activo afectado: 200.xx.xxx.xx0	
Descripción de vulnerabilidad: Según el número de versión declarado, la instalación remota de pfSense es anterior a la 2.2.6. Por lo tanto, presenta múltiples vulnerabilidades.	
Impacto vulnerabilidad: Un atacante puede provocar fallos en el firewall mediante DoS en servicios TLS.	
Mitigación: Actualizar a la versión 2.2.3 o posterior de pfSense, Usar certificados válidos y actualizados, Deshabilitar servicios de administración expuestos a Internet sin necesidad, Configurar acceso a la GUI por HTTPS con fuerte cifrado.	

Nota. *Resultados obtenidos de Nessus*

Tabla 50

Unix Operating System Unsupported Version Detection

Vulnerabilidad: Unix Operating System Unsupported Version Detection	CVE: N/A
Impacto: Critico	Riesgo: Critica
Activo afectado: 200.xx.xxx.xx0 200.xx.xxx.xx0	

Descripción de vulnerabilidad: Según el número de versión declarado, el sistema operativo Unix que se ejecuta en el host remoto ya no recibe soporte. La falta de soporte implica que el proveedor no publicará nuevos parches de seguridad para el producto. Por lo tanto, es probable que contenga vulnerabilidades de seguridad. Impacto vulnerabilidad: Cualquier CVE asociado con esa versión del SO puede ser explotado fácilmente. Servicios que se ejecutan sobre el SO (ej. OpenSSL, SSH, webGUI) también pueden estar desactualizados. Mitigación: Actualizar a la versión 2.2.3 o posterior de pfSense, Minimizar servicios expuestos (webGUI, SSH) hasta que se actualice. Habilitar firewall estricto y monitoreo de integridad del sistema.

Nota. *Resultados obtenidos de Nessus*

Tabla 51

PfSense Unsupported Version Detection

Vulnerabilidad: pfSense Unsupported Version Detection	CVE: N/A
Impacto: Critico	Riesgo: Critica
Activo afectado: 200.xx.xxx.xx0	
Descripción de vulnerabilidad: Según el número de versión autodeclarado, el host remoto pfSense ya no recibe soporte. La falta de soporte implica que el proveedor no publicará nuevos parches de seguridad para el producto. Por lo tanto, es probable que contenga vulnerabilidades de seguridad.	

Impacto vulnerabilidad: Cualquier CVE asociado con esa versión del SO puede ser explotado fácilmente. Servicios que se ejecutan sobre el SO (ej. OpenSSL, SSH, webGUI) también pueden estar desactualizados.
Mitigación: Actualizar a la versión 2.2.3 o posterior de pfSense, Minimizar servicios expuestos (webGUI, SSH) hasta que se actualice. Habilitar firewall estricto y monitoreo de integridad del sistema.

Nota. *Resultados obtenidos de Nessus*

Tabla 52

PfSense < 2.3.1-p1 Multiple Vulnerabilities (SA-16_05)

Vulnerabilidad: pfSense < 2.3.1-p1 Multiple Vulnerabilities (SA-16_05)	CVE: N/A
Impacto: Alto	Riesgo: Alto
Activo afectado: 200.xx.xxx.xx0	
Descripción de vulnerabilidad: Según el número de versión autodeclarado, la instalación remota de pfSense es anterior a la 2.3.1-p1. Por lo tanto, se ve afectada por múltiples vulnerabilidades, como se indica en los avisos de los proveedores mencionados.	
Impacto vulnerabilidad: Posible decriptación de tráfico TLS (VPN, GUI, autenticación). Riesgo de denegación de servicio (DoS) por corrupción de memoria.	
Mitigación: Actualizar inmediatamente pfSense a 2.3.1-p1 o superior. Hay que asegurar que los servicios HTTPS/VPN utilicen cifrados robustos (AES-GCM, TLS 1.2+). Restringir el acceso a la interfaz web y servicios VPN desde redes confiables.	

Nota. *Resultados obtenidos de Nessus*

Tabla 53

PfSense < 2.3.1-p5 Multiple Vulnerabilities (SA-16_07 / SA-16_08)

Vulnerabilidad: pfSense < 2.3.1-p5 Multiple Vulnerabilities (SA-16_07 / SA-16_08)	CVE: CVE-2013-7456
Impacto: Alto	Riesgo: Alto

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: Según el número de versión declarado, la instalación remota de pfSense es anterior a la 2.3.1-p5. Por lo tanto, se ve afectada por múltiples vulnerabilidades, como se indica en los avisos de los proveedores mencionados.

Impacto de vulnerabilidad: Posibilidad de denegación de servicio (DoS) al procesar tráfico malicioso, Vulnerabilidades pueden permitir ejecución de código arbitrario, Riesgo de exposición de información de red capturada.

Mitigación: Actualizar de forma imediata pfSense a 2.3.1-p1 o superior.

Limitar el uso de herramientas como tcpdump o evitar su ejecución automática en entornos productivos.

Monitorear y controlar el tráfico en interfaces críticas (WAN/DMZ).

Mantener actualizados los paquetes del sistema y dependencias (OpenSSL, libarchive, etc.).

Nota. *Resultados obtenidos de Nessus*

Tabla 54

PfSense < 2.3.1 Multiple Vulnerabilities (SA-16_03 / SA-16-04)

Vulnerabilidad: pfSense < 2.3.1 Multiple Vulnerabilities (SA-16_03 / SA-16-04)	CVE: CVE-2016-1886
Impacto: **Alto**	Riesgo: **Alto**

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: Según el número de versión declarado, la instalación remota de pfSense es anterior a la 2.3.1. Por lo tanto, presenta múltiples vulnerabilidades.

Impacto de vulnerabilidad: un atacante interno/externo puede ejecutar comandos arbitrarios en el sistema. Puede llevar a caída de servicios o uso malicioso del sistema. Puede ser usado para enviar correos fraudulentos o modificar el comportamiento del sistema de notificaciones del firewall.

Mitigación: Actualizar de forma inmediata pfSense a 2.3.1-p1 o superior.

Deshabilitar servicios innecesarios como sendmail si no son requeridos.

Monitorear y Restringir acceso administrativo y asegurar que solo usuarios confiables tengan acceso al sistema.

Monitorear logs de sistema para detectar actividades anómalas con respecto a nombres de host.

Nota. *Resultados obtenidos de Nessus.*

Tabla 55

Network Time Protocol Daemon (ntpd) Read_Mru_List() Remote DoS

Vulnerabilidad: Network Time Protocol Daemon (ntpd) read_mru_list() Remote DoS	CVE: CVE-2016-7434
Impacto: Alto	Riesgo: Medio

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: El servidor NTP remoto se ve afectado por una vulnerabilidad de denegación de servicio debido a la validación incorrecta de las consultas mrulist. Un atacante remoto no autenticado puede explotar esto mediante un paquete de consulta mrulist NTP especialmente diseñado para finalizar el proceso ntpd.

Según se informa, el servidor NTP también se ve afectado por vulnerabilidades adicionales; sin embargo, Nessus no las ha probado.

Impacto de vulnerabilidad: Puede interrumpir el servicio NTP, afectando funciones del sistema.

Mitigación: Actualizar de forma imediata pfSense a 2.3.1-p1 o superior.

Deshabilitar la opción de monitoreo remoto MRU si no es necesaria (mru en la configuración).

Filtrar el tráfico entrante UDP en el puerto 123 desde fuentes no confiables.

Nota. *Resultados obtenidos de Nessus*

Tabla 56

PfSense < 2.2.4 Multiple Vulnerabilities (SA-15_07)

Vulnerabilidad: pfSense < 2.2.4 Multiple Vulnerabilities (SA-15_07)	CVE: CVE-2015-3152
Impacto: Alto	Riesgo: Medio

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: Según el número de versión declarado, la instalación remota de pfSense es anterior a la 2.2.4. Por lo tanto, se ve afectada por múltiples vulnerabilidades, como se indica en los avisos de los proveedores mencionados.

Impacto de vulnerabilidad: Un atacante pueda explotar una condición de uso de archivos locales sin validación adecuada, dejando en

SSL Self-Signed Certificate

exposición de cookies o archivos locales que contengan información sensible. No causa denegación directa de servicio.

Mitigación: Actualizar de forma imediata pfSense a 2.3.1-p1 o superior.

Revisar configuraciones de servicios web o integraciones externas que usen libcurl.

Filtrar el Restringir el acceso administrativo solo a usuarios confiables y sobre conexiones cifradas (HTTPS/SSH).

Evitar el uso de plugins o paquetes de terceros no verificados que puedan interactuar con cookies o archivos locales.

Nota. *Resultados obtenidos de Nessus*

Tabla 57

SSL Self-Signed Certificate.

Vulnerabilidad: SSL Self-Signed Certificate.	CVE: N/A
Impacto: Medio	Riesgo: Medio

Activo afectado:

200.xx.xxx.xx0

200.xx.xxx.xx0

Descripción de vulnerabilidad: La cadena de certificados X.509 para este servicio no está firmada por una autoridad de certificación reconocida. Si el host remoto es un host público en producción, esto invalida el uso de SSL, ya que cualquiera podría establecer un ataque de intermediario contra el host remoto.

Tenga en cuenta que este complemento no comprueba las cadenas de certificados que terminan en un certificado no auto firmado, sino firmado por una autoridad de certificación no reconocida.

Impacto de vulnerabilidad: Puede permitir ataques de tipo Man-in-the-Middle (MitM) si el atacante intercepta la comunicación. Si el usuario acepta el certificado sin verificar, puede conectarse a un sistema malicioso que suplanta al firewall. NO interfiere directamente con el funcionamiento del sistema.

Mitigación: Actualizar de forma inmediata pfSense a 2.3.1-p1 o superior.

Si se usa un CA interno, importar el certificado raíz en los dispositivos de administración.

Actualizar el certificado periódicamente y verificar su validez.

Deshabilitar el uso de certificados auto firmados para acceso remoto externo.

Nota. *Resultados obtenidos de Nessus*

Tabla 58

TLS Version 1.0 Protocol Detection, TLS Version 1.1 Deprecated Protocol

Vulnerabilidad: TLS Version 1.0 Protocol Detection, TLS Version 1.1 Deprecated Protocol.	CVE: N/A
Impacto: Medio	Riesgo: Medio

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: El servicio remoto acepta conexiones cifradas con TLS 1.0. TLS 1.0 presenta varias fallas de diseño criptográfico. Las implementaciones modernas de TLS 1.0 mitigan estos problemas, pero las versiones más recientes de TLS, como la 1.2 y la 1.3, están diseñadas para evitar estas fallas y deben utilizarse siempre que sea posible.

A partir del 31 de marzo de 2020, los endpoints que no estén habilitados para TLS 1.2 y versiones posteriores dejarán de funcionar correctamente con los principales navegadores web y proveedores.
Impacto de vulnerabilidad: TLS 1.0 tiene algoritmos criptográficos débiles que pueden ser explotados para descifrar información.
Permite ataques de tipo Downgrade, facilitando la manipulación del tráfico cifrado.
No afecta directamente la operación del sistema, pero puede ser vector de ataques.
Mitigación: Actualizar de forma imediata pfSense a 2.3.1-p1 o superior.
Deshabilitar TLS 1.0 en la configuración del sistema, GUI web, OpenVPN u otros servicios.
Asegurar el soporte exclusivo de TLS 1.2 o TLS 1.3.
Revisar que los clientes o dispositivos que acceden al firewall también soporten versiones modernas.

Nota. *Resultados obtenidos de Nessus*

Tabla 59

PfSense < 2.4.2 Multiple Vulnerabilities (SA-17_07)

Vulnerabilidad: pfSense < 2.4.2 Multiple Vulnerabilities (SA-17_07)	CVE: CVE-2017-1086
Impacto: Medio	Riesgo: Medio

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: Según el número de versión autoinformado, la instalación remota de pfSense es anterior a la 2.4.2. Por lo tanto, se ve afectada por múltiples vulnerabilidades, como se indica en los avisos del proveedor.

Impacto de vulnerabilidad: Puede causar que los servicios DNS/DHCP de pfSense fallen o se reinicien.

Mitigación: Actualizar de forma imediata pfSense a 2.3.1-p1 o superior.

Limitar el acceso a los servicios DNS y DHCP desde redes no confiables.

Aplicar parches de seguridad del sistema operativo subyacente (FreeBSD) si se usa en modo personalizado.

Monitorear logs y tráfico inusual en el puerto 53 (DNS).

Nota. *Resultados obtenidos de Nessus*

Tabla 60

JQuery 1.2 < 3.5.0 Multiple XSS

Vulnerabilidad: JQuery 1.2 < 3.5.0 Multiple XSS	CVE: CVE-2020-11022
Impacto: Medio	Riesgo: Baja

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: According to the self-reported version in the script, the version of JQuery hosted on the remote web server is greater
than or equal to 1.2 and prior to 3.5.0. It is, therefore, affected by multiple cross site scripting vulnerabilities.

Note, the vulnerabilities referenced in this plugin have no security impact on PAN-OS, and/or the scenarios
required for successful exploitation do not exist on devices running a PAN-OS release.

Impacto de vulnerabilidad: No causa interrupción del sistema directamente. Un atacante podría robar cookies de sesión o credenciales del administrador.

Mitigación: Actualizar de forma imediata pfSense a 2.3.1-p1 o superior.
Restringir el acceso a la GUI web solo desde redes de administración.
Aplicar buenas prácticas de seguridad web: encabezados HTTP seguros (CSP, X-XSS-Protection), evitar exponer interfaces en WAN.
No acceder a la interfaz desde redes públicas o sin HTTPS.

Nota. *Resultados obtenidos de Nessus*

Tabla 61

OpenSSL AES-NI Padding Oracle MitM Information Disclosure.

Vulnerabilidad: OpenSSL AES-NI Padding Oracle MitM Information Disclosure.	CVE: CVE-2016-2107
Impacto: Medio	Riesgo: Baja

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: El host remoto se ve afectado por una vulnerabilidad de divulgación de información de intermediario (MitM) debido a un error en la implementación de conjuntos de cifrado que utilizan AES en modo CBC con HMAC-SHA1 o HMAC-SHA256. La implementación está diseñada específicamente para utilizar la aceleración AES disponible en procesadores x86/AMD64 (AES-NI). Los mensajes de error devueltos por el servidor permiten a un atacante intermediario realizar un ataque de oráculo de relleno, lo que permite descifrar el tráfico de red.

Impacto de vulnerabilidad: Permite a un atacante descifrar parcialmente tráfico TLS cifrado. No causa caídas ni denegaciones de servicio directamente.

Mitigación: Actualizar de forma imediata pfSense a 2.3.1-p1 o superior.
Usar VPN para el acceso remoto y evitar exponer la GUI de administración por WAN.
Deshabilitar cifrado AES CBC si es posible y usar suites TLS más modernas (como GCM).
Aplicar parches de seguridad al sistema subyacente si se usa pfSense de forma personalizada.

Nota. *Resultados obtenidos de Nessus*

Tabla 62

SSH Terrapin Prefix Truncation Weakness (CVE-2023-48795)

Vulnerabilidad: SSH Terrapin Prefix Truncation Weakness (CVE-2023-48795)	CVE: CVE-2023-48795
Impacto: Medio	Riesgo: Baja

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: El servidor SSH remoto es vulnerable a una vulnerabilidad de truncamiento de prefijo de intermediario conocida como Terrapin. Esto puede permitir que un atacante remoto de intermediario eluda las comprobaciones de integridad y reduzca la seguridad de la conexión.

Tenga en cuenta que este complemento solo busca servidores SSH remotos compatibles con ChaCha20-Poly1305 o CBC con cifrado y MAC, y no con las contramedidas estrictas de intercambio de claves. No busca versiones de software vulnerables.

Impacto de vulnerabilidad: No causa caída directa del sistema, pero permite abuso de confianza en sesiones activas. Puede permitir modificar o inyectar comandos durante la conexión SSH.

Mitigación: Actualizar de forma imediata pfSense a 2.3.1-p1 o superior.
Usar VPN para el acceso remoto y evitar exponer la GUI de administración por WAN.
Deshabilitar cifrado AES CBC si es posible y usar suites TLS más modernas (como GCM).
Aplicar parches de seguridad al sistema subyacente si se usa pfSense de forma personalizada.

Nota. *Resultados obtenidos de Nessus*

Tabla 63

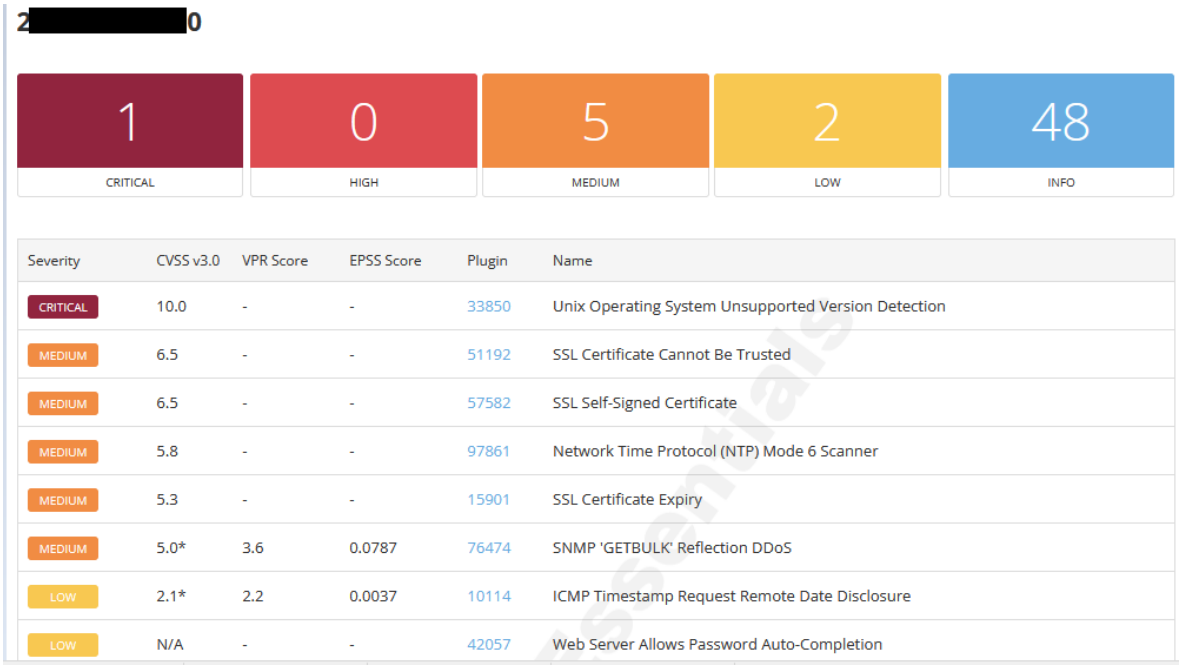
Network Time Protocol (NTP) Mode 6 Scanner

Vulnerabilidad: Network Time Protocol (NTP) Mode 6 Scanner.	CVE: CVE-2023-48795
Impacto: Medio	Riesgo: Baja
Activo afectado: 200.xx.xxx.xx0 200.xx.xxx.xx0	
Descripción de vulnerabilidad: El servidor NTP remoto responde a consultas de modo 6. Los dispositivos que responden a estas consultas podrían utilizarse en ataques de amplificación de NTP. Un atacante remoto no autenticado podría aprovechar esto, mediante una consulta de modo 6 especialmente diseñada, para provocar una denegación de servicio reflejada.	
Impacto de vulnerabilidad: Filtración de información del sistema, posible abuso para ataques DDoS reflejados.	
Mitigación: Actualizar de forma imediata pfSense a 2.3.1-p1 o superior. Restringir acceso NTP, usar noquery, bloquear modo 6.	

Nota. *Resultados obtenidos de Nessus*

Figura 69

Escaneo 2x0.2xx.xx.xxx



Nota. Captura de pantalla de resultado generado por Nessus

Tabla 64

SSL Certificate Expiry.

Vulnerabilidad: SSL Certificate Expiry.	CVE: N/A
Impacto: Medio	Riesgo: Baja
Activo afectado: 200.xx.xxx.xx0 200.xx.xxx.xx0	
Descripción de vulnerabilidad: Este complemento verifica las fechas de vencimiento de los certificados asociados a servicios con SSL habilitado en el destino e informa si alguno ya ha vencido.	

Impacto de vulnerabilidad: Genera alertas de seguridad en navegadores, errores en clientes VPN, interfaces de administración, etc. usuarios pueden no acceder a la interfaz web o APIs si sus navegadores o herramientas bloquean el acceso.
Mitigación: Actualizar de forma inmediata pfSense a 2.3.1-p1 o superior.
Compre o genere un nuevo certificado SSL para reemplazar el existente.

Nota. *Resultados obtenidos de Nessus*

Tabla 65

SNMP 'GETBULK' Reflection DDoS.

Vulnerabilidad: SNMP 'GETBULK' Reflection DDoS.	CVE: CVE-2008-4309
Impacto: Medio	Riesgo: Media
Activo afectado:	
200.xx.xxx.xx0	
200.xx.xxx.xx0	
Descripción de vulnerabilidad: El demonio SNMP remoto responde con una gran cantidad de datos a una solicitud 'GETBULK' con un valor de 'max-repetitions' mayor de lo normal. Un atacante remoto puede usar este servidor SNMP para realizar un ataque de denegación de servicio distribuido reflejado en un host remoto arbitrario.	
Impacto de vulnerabilidad: Puede saturar el ancho de banda saliente de tu red. Participar en ataques reflejados puede incluirte en listas negras.Puede ser usado como un nodo reflector en ataques DDoS.	
Mitigación: Actualizar de forma imediata pfSense a 2.3.1-p1 o superior.	
Restringir acceso SNMP solo a IPs autorizadas.	
Evitar exponer SNMP a Internet público	

Nota. *Resultados obtenidos de Nessus*

Tabla 66

Web Server Allows Password Auto-Completion.

Vulnerabilidad: Web Server Allows Password Auto-Completion.	CVE: N/A
Impacto: Baja	Riesgo: Baja

Activo afectado: 200.xx.xxx.xx0

Descripción de vulnerabilidad: El servidor web remoto contiene al menos un campo de formulario HTML con una entrada de tipo "contraseña" donde la función de autocompletar no está desactivada.

Si bien esto no representa un riesgo para el servidor web en sí, sí implica que los usuarios que utilizan los formularios afectados podrían tener sus credenciales guardadas en sus navegadores, lo que podría provocar una pérdida de confidencialidad si alguno de ellos utiliza un servidor compartido o si su equipo se ve comprometido en algún momento.

Impacto de vulnerabilidad: No afecta el acceso al sistema. Si otro usuario accede al navegador, puede ver o usar credenciales almacenadas.

Mitigación: No guardar contraseñas en navegadores en equipos compartidos o no protegidos, Usar administradores de contraseñas seguros.

Nota. *Resultados obtenidos de Nessus*

Medidas y acciones para fortalecer la seguridad de la infraestructura del centro de datos de la compañía

Tabla 67

Acciones para Fortalecer la Seguridad en la Infraestructura

Criticidad	Activo Afectado	Vulnerabilidad Principal	Recomendación	Tiempo Estimado
Crítica	Firewall pfSense interno 1	Firewalls que ejecutan versiones antiguas de pfSense (con vulnerabilidades críticas como CVE-2019-12462) deben actualizarse a versiones soportadas (2.4.5 o superior).	Migrar a versión actual de pfSense a versiones soportadas (2.4.5 o superior).	1-2 semanas
Crítica	Firewall pfSense interno 2	Exposición de interfaz web sin autenticación robusta (CVE 2024...)	Implementar autenticación multifactor y restringir acceso por IP.	1 semana
Alta	Switch Huawei de distribución	Vulnerabilidad DoS (CVE-2014-4190, CVE-2014-4705 y CVE-2014-4707)	Actualizar firmware del switch a la versión recomendada por el fabricante.	1 semana
Media	Varios dispositivos (9)	Servicio Telnet habilitado (sin cifrado)	Deshabilitar Telnet. Implementar y forzar uso de SSH con autenticación por clave para evitar interceptación de credenciales.	1 semana

Media	Varios dispositivos	Certificados SSL no confiables	Adquirir certificados válidos de AC reconocida.	2 semanas
Baja	Varios dispositivos	Respuesta a solicitudes ICMP Timestamp (CVE-1999-0524)	Bloquear tipos ICMP 13 y 14 en el firewall.	1-2 días

Nota. La siguiente tabla representa las *Acciones para Fortalecer la Seguridad en la Infraestructura*

Endurecimiento de la Seguridad

- Configuración de reglas estrictas en firewall para el bloqueo de tráfico no autorizado (paquetes maliciosos en puertos UDP/SNMP).
- Implementación de cifrado robusto (TLS 1.2/1.3) en servicios web y VPN, reemplazando certificados auto firmados por certificados validos ante entidad reconocidas.
- Validar la segmentación de redes por medio de VLANs y reglas de Firewall para limitar el movimiento lateral en caso de compromiso.

Auditorías Periódicas y Pruebas de Penetración

Se recomienda establecer un plan de escaneos de vulnerabilidades trimestrales con herramientas como Nessus o OpenVAS para identificar nuevos riesgos.

Se recomienda complementar con pruebas de penetración anuales (Pentesting Caja Negra), con el fin de simular ataques externos y validar efectividad de las medidas implementadas.

Conclusiones

El proyecto de grado aplicado “Análisis y Evaluación de Vulnerabilidades en Infraestructura de Seguridad en un Centro de Datos” permitió alcanzar los objetivos planteados mediante la ejecución de un análisis exhaustivo y estructurado, logrando identificar múltiples vulnerabilidades críticas, altas, medias y bajas que podrían comprometer la confidencialidad, integridad y disponibilidad de los activos tecnológicos más sensibles de la empresa.

La adopción de la metodología PTES al proyecto, demostró ser efectivo para organizar las fases de recolección de la información, modelado de amenazas, análisis de vulnerabilidades y documentación de los hallazgos. La implementación de la técnica Caja blanca es la más adecuada en entornos de producción, permitiendo un análisis profundo sin poner en riesgo la operación de los sistemas.

La clasificación y priorización de los activos permitió concentrar los esfuerzos sobre los activos más críticos, logrando visibilidad sobre routers, firewall y switches que requieren validación inmediata. Asimismo, la correlación de amenazas mediante el uso de la herramienta Microsoft Threat Modeling Toll arrojo un panorama detallado de los riesgos potenciales y posibles consecuencias operativas.

La implementación de scripts en Python, integrados con herramientas como Shodan, Scapy y el módulo socket, optimizó las fases de recopilación de información y escaneo de puertos. Esta automatización no solo redujo los tiempos de análisis y minimizó errores humanos, sino que también facilitó la repetibilidad del proceso, asegurando consistencia en futuras auditorías.

El análisis realizado con Nessus arrojo un resultado de 70 vulnerabilidades. Entre los principales hallazgos destaca la presencia de protocolos inseguros como telnet (Unencrypted

Telnet Server) en 9 activos, Firewalls con sistemas operativos obsoletos (p. ej., pfSense) y vulnerabilidades críticas asociadas a versiones sin soporte, como CVE-2019-12462, que permiten ejecución remota de código e infraestructuras que aún operan con sistemas sin soporte o sin parches de seguridad aplicados.

Los resultados obtenidos en la auditoria dejan en evidencia la necesidad de actualizar sistemas obsoletos, reemplazar protocolos inseguros (Telnet por SSH v2) y endurecer las políticas de acceso. La documentación detallada de los hallazgos brindo a la empresa proveedora de servicios de Internet (ISP). una sólida base para implementar un plan de acción PDA correctivas, como la segmentación de redes, la implementación de MFA y monitorización continua.

Recomendaciones

Actualizar los sistemas pfSense obsoletos a versiones estables y soportadas por fabrica, con el fin de mitigar múltiples vulnerabilidades detectadas en versiones antiguas.

Fortalecer mecanismos de autenticación: mediante la implementación de autenticación multifactor (MFA) y la restricción de accesos administrativos por direcciones IP autorizadas.

Actualización de firmware de los dispositivos de conmutación a versiones recomendadas por el fabricante, con el fin de corregir vulnerabilidades conocidas y garantizar la estabilidad del sistema.

Implementar accesos remotos seguros: hacia los dispositivos de red y seguridad, utilizando protocolos cifrados como SSH y deshabilitando protocolos inseguros como Telnet.

Aplicar políticas de gestión de certificados digitales, asegurando el uso de certificados confiables, vigentes y emitidos por entidades certificadoras reconocidas.

Limitar la divulgación de servicios y banners, configurando los equipos para reducir la exposición de versiones de software y detalles del sistema.

Endurecer la configuración de servicio SNMP, limitando el acceso por listas de control, usando versiones seguras y evitando configuraciones por defecto.

Establecer una política formal para la gestión de parches, que permita actualización constante de sistemas operativos. Dispositivos de red y plataformas de seguridad.

Implementar un plan de auditorías periódicas, implementando herramientas de escaneo de vulnerabilidades, para evaluar la efectividad de medidas correctivas aplicadas.

Documentación y estandarización de configuraciones seguras (hardinng) para los dispositivos críticos del centro de datos, alineados con buenas prácticas y estándares internacionales.

Referencias Bibliográficas

Anaconda, Inc. (2023). *Conda*. https://docs.conda.io/

Amazon Web Services, Inc. (s. f.). *¿Qué es un centro de datos? - Explicación de los centros de datos en la nube - AWS*. Recuperado el 10 de julio de 2025, de https://aws.amazon.com/es/what-is/data-center/

Center for Internet Security. (2021). *CIS Critical Security Controls v8*. https://www.cisecurity.org/controls/v8

CVSS v4.0 Specification Document. (s. f.). *FIRST — Forum Of Incident Response And Security Teams*. Recuperado el 1 de mayo de 2025, de https://www.first.org/cvss/specification-document

Fortinet. (s. f.). *¿Qué es un firewall de red? Proteja la red de tráfico*. Recuperado el 1 de julio de 2025, de https://www.fortinet.com/lat/resources/cyberglossary/firewall

Gianni. (2023, 28 de julio). *LACNIC impulsa creación de centros de respuesta a seguridad informática*. LACNIC Blog. https://blog.lacnic.net/lacnic-impulsa-creacion-de-centros-de-respuesta-a-seguridad-informatica/

Hackers de sombrero negro, blanco y gris: definición y explicación. (2023, 17 de agosto). *Kaspersky Latinoamérica*. https://latam.kaspersky.com/resource-center/definitions/hacker-hat-types

Incibe. (s. f.). *Amenaza vs Vulnerabilidad, ¿sabes en qué se diferencian? | Empresas | INCIBE*. Recuperado el 11 de julio de 2025, de https://www.incibe.es/empresas/blog/amenaza-vs-vulnerabilidad-sabes-se-diferencian

ISACA. (2019). *COBIT 2019 Framework: Governance and Management Objectives*. ISACA.

ISO/IEC. (2013). *ISO/IEC 27001:2013 – Information Security Management Systems – Requirements*. International Organization for Standardization.

ISO/IEC. (2022). *ISO/IEC 27002:2022 – Information security, cybersecurity and privacy protection – Information security controls*. International Organization for Standardization.

Kali Linux. (s. f.). *What is Kali Linux? | Kali Linux Documentation*. Recuperado el 1 de junio de 2025, de https://www.kali.org/docs/introduction/what-is-kali-linux/

Kim, D., & Solomon, M. G. (2016). *Fundamentals of Information Systems Security* (3rd ed.). Jones & Bartlett Learning.

Lyon, G. F. (2009). *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure.Com LLC.

Maynor, D. (2011). *Metasploit Toolkit for Penetration Testing, Exploit Development, and Vulnerability Research* (2nd ed.). Syngress.

NIST. (2008). *Technical Guide to Information Security Testing and Assessment (SP 800-115)*. https://csrc.nist.gov/publications/detail/sp/800-115/final

NIST. (2020). *Framework for Improving Critical Infrastructure Cybersecurity* (Version 1.1). National Institute of Standards and Technology.

OWASP. (2021). *OWASP Top 10:2021*. https://owasp.org/www-project-top-ten/

Pankov, N. (2021, 17 de mayo). *El ataque de ransomware a Colonial Pipeline*. Kaspersky. https://www.kaspersky.es/blog/pipeline-ransomware-mitigation/25302/

Peltier, T. R. (2016). *Information Security Policies, Procedures, and Standards: Guidelines for Effective Information Security Management*. Auerbach Publications.

Peltier, T. R. (2016). *Information Security Policies, Procedures, and Standards*. CRC Press.

Pre-engagement - the Penetration Testing Execution Standard. (s. f.). http://www.pentest-standard.org/index.php/Pre-engagement

Python Software Foundation. (2023). *Python (versión 3.11)*. https://www.python.org/

Scarfone, K., & Mell, P. (2007). *Guide to Vulnerability Assessment (NIST SP 800-115)*. National Institute of Standards and Technology.

Shodan. (2023). *Shodan: The search engine for the Internet of Things*. https://www.shodan.io/

Shostack, A. (2014). *Threat Modeling: Designing for Security*. Wiley.

TCP SYN (Stealth) Scan (-SS) | NMAP Network Scanning. (s. f.).

Tenable Once Again Named One Of The Top 20 Cloud Security Companies By CRN. (2025).

Vulnerability Analysis - The Penetration Testing Execution Standard. (s. f.).

ACIS. (s. f.). *Conozca los Principales Desafíos de Seguridad Digital Que Tiene Colombia Para el 2024*. https://www.acis.org.co/

Scapy. (s. f.). *Introduction — Scapy 2.6.1 documentation*. Recuperado el 12 de junio de 2025, de https://scapy.readthedocs.io/en/latest/introduction.html

Nmap. (s. f.). *Guía de Referencia de Nmap (Página de Manual)*. Recuperado el 15 de junio de 2025, de https://nmap.org/man/es/man-bypass-firewalls-ids.html

Apéndices

Apéndice A

Código fuente: Análisis y Evaluación de Vulnerabilidades en Infraestructura de Seguridad

```
##Importante: Este Código requiere la instalación de las librerías scapy, rich, tqdm y graphviz.


import ipaddress
import logging
from concurrent.futures import ThreadPoolExecutor
from rich.console import Console
from rich.table import Table
from tqdm import tqdm
from scapy.layers.inet import IP, TCP
from scapy.all import *
from graphviz import Digraph


logging.getLogger("scapy.runtime").setLevel(logging.ERROR)


class Red_Datacenter_seguridad:
    def __init__(self, network_range, timeout=1):
        self.network_range = network_range
        self.timeout = timeout


    def _scan_host_sockets(self, ip, port=1000):
        try:
            with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
```

```
                s.settimeout(self.timeout)
                s.connect((ip, port))
                return (port, True)
        except (socket.timeout, socket.error):
            return (port, False)

    def _scan_host_scapy(self, ip, scan_ports=(80, 4443, 8443, 23)):
        for port in scan_ports:
            packet = IP(dst=ip, ttl=64)/TCP(dport=port, flags='S', window=0x4001,
options=[('MSS', 1460)])
            respuesta, _ = sr(packet, timeout=self.timeout, verbose=0)
            if respuesta:
                return (port, True)
        return (port, False)

    def scan_specific_hosts(self, hosts_list, scan_ports=(80, 4443, 8443, 23)):
        hosts_up = []
        with ThreadPoolExecutor(max_workers=100) as executor:
            futures = {
                executor.submit(self._scan_host_scapy, ip, scan_ports): ip
                for ip in tqdm(hosts_list, desc="Escaneando IPs específicas")
            }
            for future in tqdm(futures, desc="Obteniendo resultados"):
                if future.result()[1]:
                    hosts_up.append(future.result()[0])
        return hosts_up
```

```
    def ports_scan(self, hosts_list, port_range=(1, 1000), extra_ports=(4443, 8443,
23)):
        all_open_ports = {}
        for ip in hosts_list:
            open_ports = []
            for port in tqdm(range(*port_range), desc=f"Escaneando puertos en {ip}"):
                try:
                    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
                        s.settimeout(self.timeout)
                        result = s.connect_ex((ip, port))
                        if result == 0:
                            open_ports.append(port)
                except Exception:
                    continue
            if open_ports:
                all_open_ports[ip] = open_ports
        return all_open_ports


    def graficar_puertos_abiertos(resultados):
        dot = Digraph(comment='Puertos abiertos por IP')
        for ip, puertos in resultados.items():
            dot.node(ip)
            for puerto in puertos:
                dot.node(f"{ip}:{puerto}", label=str(puerto), shape='box')
                dot.edge(ip, f"{ip}:{puerto}")
        dot.render('puertos_abiertos', format='png', cleanup=True)


    def get_banner(self, ip, port):
```

```
        try:
            with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
                s.settimeout(self.timeout + 20)
                s.connect((ip, port))
                s.send(b'Hello\r\n')
                return s.recv(6144).decode(errors="ignore").strip()
        except Exception as e:
            return f"Error: {e}"

    def services_scan(self, hosts_ports_dict):
        services_info = {}
        with ThreadPoolExecutor(max_workers=100) as executor:
            for ip, ports in hosts_ports_dict.items():
                futures = []
                services_info[ip] = {}
                for port in tqdm(ports, desc=f"Obteniendo banners en {ip}"):
                    future = executor.submit(self.get_banner, ip, port)
                    futures.append((future, port))
                for future, port in futures:
                    result = future.result()
                    if result and 'timed out' not in result and 'refused' not in
result and 'No route to host' not in result:
                        services_info[ip][port] = result
        return services_info

    def pretty_print(self, data, data_type="hosts"):
        console = Console()
        table = Table(show_header=True, header_style="bold red1")
```

```
if data_type == "hosts":
    table.add_column("Hosts Activos Datacenter", style="chartreuse1")
    for host in data:
        table.add_row(host, end_section=True)
elif data_type == "ports":
    table.add_column("Dirección IP", style="bright_white on blue3")
    table.add_column("Puertos Abiertos", style="chartreuse1")
    for ip, ports in data.items():
        ports_str = ", ".join(map(str, ports))
        table.add_row(ip, ports_str, end_section=True)
elif data_type == "services":
    table.add_column("IP Address", style="bold green")
    table.add_column("Port", style="bold blue")
    table.add_column("Service", style="bold yellow")
    for ip, services in data.items():
        for port, service in services.items():
            table.add_row(ip, str(port), service, end_section=True)
console.print(table)
```

Apéndices B

Código fuente: Análisis y Evaluación de Vulnerabilidades en Infraestructura de Seguridad

```
from network_analyzer import Red_Datacenter_seguridad

# Leer IPs desde archivo
with open("/home/*******k/Escritorio/Recopilación_Activa/host_objetivo.txt", "r") as
file:
    scan_specific_hosts = [line.strip() for line in file if line.strip()]

# Crear instancia del analizador
analyzer = Red_Datacenter_seguridad('192.168.1.0/24')

# Escanear los puertos del 1 al 1000 y también 4443 y 8443
resultados = analyzer.ports_scan(
   hosts_list=scan_specific_hosts,
    port_range=(1, 1001),
    extra_ports=(4443, 8443)
)
services = analyzer.services_scan(resultados)

# Mostrar resultados
analyzer.pretty_print(services, data_type="services")
```

Apéndice C

Glosario

Amenaza informática: acción o evento dirigidas a explotación de vulnerabilidades de un sistema de información que pueda comprometer la confidencialidad, disponibilidad e integridad de la información. Las amenazas informáticas pueden ser internas, externas o accidentales.

Vulnerabilidad informática: Debilidad o fallo en el diseño, implementación, configuración o gestión de sistemas, aplicaciones o redes que pueden ser explotadas por una amenaza para ganar acceso y comprometer la seguridad.

Seguridad de la información (InfoSeg): Conjunto de medidas y herramientas de seguridad destinadas a proteger la información confidencial de la empresa contra accesos no autorizados, interrupciones, modificaciones, divulgación o destrucción. Está basado en principios de confidencialidad, integridad y disponibilidad (CID).

Seguridad informática (Ciberseguridad): Se especializa en la protección de sistemas, redes, programas y datos contra ataques digitales. Combina técnica de defensa, detección y respuesta ante incidentes.

Testing Caja Blanca (White Box): Técnica de penetración donde el evaluador tiene acceso completo a la información interna del sistema, como arquitectura de red, código fuente, configuraciones y documentación. Esta técnica permite un análisis profundo y detallado.

Testing Caja Gris (Gray Box): Esta técnica tiene un enfoque intermedio, donde el evaluador tiene conocimiento parcial del sistema. Combina técnica de caja blanca y caja negra para simulación de ataque con información limitada.

Testing Caja Negra (Black Box): Técnica donde el evaluador no cuenta con conocimiento previo de la infraestructura o código fuente que se está analizando. El objetivo principal es simular un ataque externo desde la perspectiva de un ciberdelincuente sin información interna.

Hacker: Persona experta en informática que puede utilizar sus habilidades para a explotación de redes o sistemas de computación con fines ilícitos o para mejorar la seguridad y se clasifican en:

Sombrero Blanco (White Hat): Expertos en seguridad que actúan bajo permiso en busca de vulnerabilidades para ayudar a corregirlas.

Sombrero Negro (Black Hat): Ciberdelincuentes que exploran y explotan vulnerabilidades con fines maliciosos buscando compensación económica o sabotaje.

Sombrero Gris (Gray Hat): Actuan entre los dos grupos anteriores, operan sin autorización, pero sin intenciones maliciosas.

Python: Es un lenguaje de programación interpretado, de alto nivel y multipropósito, se caracteriza por su sintaxis clara y legible. Ampliamente utilizado en ciberseguridad para la automatización de tareas, desarrollo de herramientas y análisis de datos.

Riesgo informático: Probabilidad de que una amenaza explote una vulnerabilidad, causando un impacto negativo en los activos de información. Se calcula con la siguiente formula:

Riesgo= Amenaza x Vulnerabilidad x Impacto.

Firewall: Dispositivo o software de red que actúa como una barrera para controlar el tráfico de red entre redes seguras y no seguras mediante la configuración de reglas predefinidas. Los firewalls pueden ser de red, aplicaciones o próxima generación (NGFW).